

# Navicat<sup>TM</sup>

User Manual

## Table of Contents

<b>WELCOME TO NAVICAT!</b>	<b>23</b>
SYSTEM REQUIREMENTS	25
FEATURE MATRIX	26
<i>Navicat Premium</i>	27
<i>Navicat for MySQL</i>	33
<i>Navicat for Oracle</i>	38
<i>Navicat for PostgreSQL</i>	43
<i>Navicat for SQLite</i>	48
<i>Navicat for SQL Server</i>	53
REGISTRATION	57
INSTALLATION	58
MAINTENANCE/UPGRADE	60
END-USER LICENSE AGREEMENT	61
<b>GETTING STARTED</b>	<b>69</b>
CONNECTION SETTINGS	71
<i>General Settings for MySQL</i>	74
<i>General Settings for Oracle</i>	75
Oracle Basic Connection General Settings	76
Oracle TNS Connection General Settings	77
<i>General Settings for PostgreSQL</i>	78
<i>General Settings for SQLite</i>	79
<i>General Settings for SQL Server</i>	80
<i>SSH Settings (Available only for MySQL, Oracle, PostgreSQL and SQL Server and supports SSH2 Protocol only)</i>	81
Benefit of SSH Tunneling	82
Password Authentication	83
Public Key Authentication	85
<i>HTTP Settings (Available only for MySQL, PostgreSQL and SQLite)</i>	87
<i>SSL Settings (Available only for MySQL and PostgreSQL)</i>	88
Installation of OpenSSL and MySQL/PostgreSQL	89
Setting up SSL Certificate for MySQL/PostgreSQL	90
Setting up Client Certificate for Navicat	93
<i>Advanced Settings</i>	95
Setting Advanced Database Properties (Available only for MySQL and PostgreSQL)	99
WORKING WITH DATABASES OR SCHEMAS	100
<i>MySQL Database Management</i>	101

<i>Oracle Schema Management</i>	<i>103</i>
<i>PostgreSQL Database Management</i>	<i>104</i>
PostgreSQL Database Designer	106
Editing PostgreSQL Database General	107
<i>SQLite Database Management</i>	<i>108</i>
<i>SQL Server Database Management</i>	<i>110</i>
SQL Server Database Designer	111
Editing SQL Server Database General	112
Editing SQL Server Database Filegroups	114
Editing SQL Server Database Files	115
Editing Advanced SQL Server Database Properties	116
WORKING WITH DATABASE OR SCHEMA OBJECTS	122
<i>Viewing Object Information</i>	<i>123</i>
<b>DATABASE OBJECT MANAGEMENT</b>	<b>124</b>
<b>MYSQL DATABASE OBJECT MANAGEMENT</b>	<b>125</b>
MYSQL TABLES	126
<i>MySQL Table Designer</i>	<i>131</i>
MySQL Table Fields	132
Setting MySQL Table Field Properties	134
Setting MySQL Table Other Field Properties	136
MySQL Table Indexes	138
Setting MySQL Table Index Properties	139
MySQL Table Foreign Keys	141
Setting MySQL Table Foreign Key Properties	143
MySQL Table Triggers	144
Setting MySQL Table Trigger Properties	145
MySQL Table Options	146
Setting MySQL Table Partition Options	149
MYSQL VIEWS	151
<i>MySQL View Designer</i>	<i>155</i>
Working with MySQL View Builder (Available only in Full Version)	156
Editing MySQL View SQL Definition	157
Setting Advanced MySQL View Properties	158
MySQL View Preview	160
MySQL View Explain	161
<i>MySQL View Viewer</i>	<i>162</i>
MYSQL FUNCTIONS/PROCEDURES	163

<i>MySQL Function Wizard</i>	166
Setting MySQL Routine Type	167
Setting Parameters for MySQL Procedure/Function	168
Setting Return Type for MySQL Function	169
<i>MySQL Function/Procedure Designer</i>	170
Editing MySQL Function/Procedure Definition	171
Setting Advanced MySQL Function/Procedure Properties	172
Viewing MySQL Function/Procedure Result	173
MySQL EVENTS	174
<i>MySQL Event Designer</i>	177
Editing MySQL Event Definition	178
Setting Advanced MySQL Event Properties	179
<b>ORACLE DATABASE OBJECT MANAGEMENT</b>	<b>181</b>
ORACLE DEBUGGER (AVAILABLE ONLY IN FULL VERSION)	182
<i>Oracle Debugger Code Window</i>	183
<i>Oracle Debugger Views</i>	184
Repositioning Oracle Debugger Views	185
Oracle Debugger Breakpoints	186
Oracle Debugger Call Stack	187
Oracle Debugger DBMS Output	188
Oracle Debugger Log	189
Oracle Debugger Smart Data	190
Oracle Debugger Watch List	191
<i>Oracle Debugger Toolbar</i>	192
EDITING ORACLE PHYSICAL ATTRIBUTES/DEFAULT STORAGE CHARACTERISTICS	193
ORACLE TABLES	196
<i>Oracle Normal Tables</i>	201
Table Designer for Oracle Normal Tables (/Index Organized Tables)	202
Oracle Table Fields	203
Setting Oracle Table Field Properties	205
Setting Other Oracle Table Field Properties	208
Oracle Table Indexes	210
Setting Oracle Table Index Properties	212
Oracle Table Foreign Keys	214
Setting Oracle Table Foreign Key Properties	215
Oracle Table Uniques	216
Oracle Table Setting Unique Properties	217
Oracle Table Checks	218



Setting Oracle Table Check Properties	219
Oracle Table Triggers	220
Setting Oracle Table Trigger Properties	221
Oracle Table Options	223
<i>Oracle External Tables</i>	224
Table Designer for Oracle External Tables	225
Fields for Oracle External Tables	226
Setting Field Properties for Oracle External Tables	228
Setting Other Field Properties for Oracle External Tables	231
External Properties for Oracle External Tables	232
Access Parameters for Oracle External Tables	233
<i>Oracle Index Organized Tables</i>	234
Table Designer for Oracle Index Organized Tables	235
Options for Oracle Index Organized Tables	236
ORACLE VIEWS	238
<i>Oracle View Designer</i>	242
Working with Oracle View Builder (Available only in Full Version)	243
Editing Oracle View SQL Definition	244
Setting Advanced Oracle View Properties	245
Oracle View Preview	246
Oracle View Explain	247
<i>Oracle View Viewer</i>	248
ORACLE FUNCTIONS/PROCEDURES	250
<i>Oracle Function Wizard</i>	254
Setting Oracle Routine Type	255
Setting Parameters for Oracle Procedure/Function	256
Setting Return Type for Oracle Function	257
<i>Oracle Function/Procedure Designer</i>	258
Editing Oracle Function/Procedure Definition	259
Viewing Oracle Function/Procedure Result	260
Debugging Oracle Function/Procedure (Available only in Full Version)	261
ORACLE DATABASE LINKS	262
<i>Oracle Database Link Designer</i>	264
Edit Oracle Database Link General	265
ORACLE INDEXES	266
<i>Oracle Index Designer</i>	268
Editing Oracle Index General	269
Oracle Normal Index	270

Oracle Unique Index	271
Oracle Bitmap Index	272
Oracle Domain Index	273
Oracle Cluster Index	274
Editing Advanced Oracle Index Properties	275
ORACLE JAVA	277
<i>Oracle Java Designer</i>	279
Editing Oracle Java General	280
Oracle Java Source	281
Oracle Java Class	282
Oracle Java Resource	283
Setting Advanced Oracle Java Properties	284
ORACLE MATERIALIZED VIEWS	285
<i>Oracle Materialized View Designer</i>	288
Working with Oracle Materialized View Builder (Available only in Full Version)	289
Editing Oracle Materialized View SQL Definition	290
Setting Advanced Oracle Materialized View Properties	291
<i>Oracle Materialized View Preview</i>	295
<i>Oracle Materialized View Explain</i>	296
ORACLE MATERIALIZED VIEW LOGS	297
<i>Oracle Materialized View Log Designer</i>	300
Editing Oracle Materialized View Log General	301
ORACLE PACKAGES	303
<i>Oracle Package Designer</i>	306
Editing Oracle Package Definition	307
<i>Oracle Package Body Designer</i>	308
Editing Oracle Package Body Definition	309
<i>Viewing Oracle Package Result</i>	310
<i>Debugging Oracle Package (Available only in Full Version)</i>	311
ORACLE SEQUENCES	312
<i>Oracle Sequence Designer</i>	314
Editing Oracle Sequence General	315
ORACLE SYNONYMS	316
<i>Oracle Synonym Designer</i>	318
Editing Oracle Synonym General	319
ORACLE TRIGGERS	320
<i>Oracle Trigger Designer</i>	323
Editing Oracle Trigger General	324

Oracle Table Trigger	325
Oracle View Trigger	327
Oracle Schema Trigger	328
Oracle Database Trigger	329
Setting Advanced Oracle Trigger Properties	330
Editing Oracle Trigger Definition	331
ORACLE TYPES	332
<i>Oracle Object Type Designer</i>	336
Editing Oracle Object Type Definition	337
<i>Oracle Type Body Designer</i>	338
Editing Oracle Type Body Definition	339
<i>Oracle Collection Type Designer</i>	340
Editing Oracle Collection Type General	341
ORACLE XML SCHEMAS	342
<i>Oracle XML Schema Designer</i>	344
Editing Oracle XML Schema Doc	345
Setting Advanced Oracle XML Schema Properties	346
ORACLE RECYCLE BIN	348
ORACLE DIRECTORIES	350
<i>Oracle Directory Designer</i>	352
Editing Oracle Directory General	353
ORACLE TABLESPACES	354
<i>Oracle Tablespace Designer</i>	356
Editing Oracle Tablespace General	357
Editing Oracle Tablespace Storage	359
Setting Advanced Oracle Tablespace Properties	361
ORACLE PUBLIC DATABASE LINKS	362
ORACLE PUBLIC SYNONYMS	363
<b>POSTGRESQL DATABASE OBJECT MANAGEMENT</b>	<b>364</b>
POSTGRESQL SCHEMAS	365
<i>PostgreSQL Schema Designer</i>	367
Editing PostgreSQL Schema General	368
<i>PostgreSQL Tables</i>	369
PostgreSQL Table Designer	374
PostgreSQL Table Fields	375
Setting PostgreSQL Table Field Properties	377
Setting Other PostgreSQL Table Field Properties	380
PostgreSQL Table Indexes	381

Setting PostgreSQL Table Index Properties	382
PostgreSQL Table Foreign Keys	383
Setting PostgreSQL Table Foreign Key Properties	384
PostgreSQL Table Uniques	385
Setting PostgreSQL Table Unique Properties	386
PostgreSQL Table Checks	387
Setting PostgreSQL Table Check Properties	388
PostgreSQL Table Excludes	389
Setting PostgreSQL Table Exclude Properties	390
PostgreSQL Table Rules	391
Setting PostgreSQL Table Rule Properties	392
PostgreSQL Table Triggers	393
Setting PostgreSQL Table Trigger Properties	394
PostgreSQL Table Options	396
<i>PostgreSQL Views</i>	<b>397</b>
PostgreSQL View Designer	401
Working with PostgreSQL View Builder (Available only in Full Version)	402
Editing PostgreSQL View SQL Definition	403
Setting Advanced PostgreSQL View Properties	404
PostgreSQL View Preview	405
PostgreSQL View Explain	406
PostgreSQL View Viewer	407
<i>PostgreSQL Functions</i>	<b>408</b>
PostgreSQL Function Wizard	412
Setting Parameters for PostgreSQL Function	413
Setting Return Type for PostgreSQL Function	414
PostgreSQL Function Designer	415
Editing PostgreSQL Function Definition	416
Setting Advanced PostgreSQL Function Properties	417
Viewing PostgreSQL Function Result	419
<i>PostgreSQL Aggregates</i>	<b>420</b>
PostgreSQL Aggregate Designer	422
Editing PostgreSQL Aggregate Properties	423
<i>PostgreSQL Conversions</i>	<b>424</b>
PostgreSQL Conversion Designer	426
Editing PostgreSQL Conversion Properties	427
<i>PostgreSQL Domains</i>	<b>428</b>
PostgreSQL Domain Designer	430

Editing PostgreSQL Domain General	431
Editing PostgreSQL Domain Check	432
<i>PostgreSQL Trigger Functions</i>	<i>433</i>
PostgreSQL Trigger Function Designer	435
Editing PostgreSQL Trigger Function Definition	436
Setting Advanced PostgreSQL Trigger Function Properties	437
<i>PostgreSQL Operators</i>	<i>439</i>
PostgreSQL Operator Designer	441
Editing PostgreSQL Operator General	442
Editing Advanced PostgreSQL Operator Properties	443
<i>PostgreSQL Operator Classes</i>	<i>444</i>
PostgreSQL Operator Class Designer	446
Editing PostgreSQL Operator Class General	447
Editing PostgreSQL Operator Class Operators	448
Editing PostgreSQL Operator Class Functions	449
<i>PostgreSQL Sequences</i>	<i>450</i>
PostgreSQL Sequence Designer	452
Editing PostgreSQL Sequence General	453
<i>PostgreSQL Types</i>	<i>455</i>
PostgreSQL Type Designer	457
Editing PostgreSQL Base Type Properties	458
Editing PostgreSQL Base Type General	459
Editing Advanced PostgreSQL Base Type Properties	460
Editing PostgreSQL Composite Type Properties	461
Editing PostgreSQL Composite Type General	462
Editing PostgreSQL Enum Type Properties	463
Editing PostgreSQL Enum Type General	464
POSTGRESQL TABLESPACES	465
<i>PostgreSQL Tablespace Designer</i>	<i>467</i>
Editing PostgreSQL Tablespace General	468
POSTGRESQL CASTS	469
<i>PostgreSQL Cast Designer</i>	<i>471</i>
Editing PostgreSQL Cast General	472
POSTGRESQL LANGUAGES	473
<i>PostgreSQL Language Designer</i>	<i>475</i>
Editing PostgreSQL Language General	476
<b>SQLITE DATABASE OBJECT MANAGEMENT</b>	<b>477</b>
SQLITE TABLES	478

<i>SQLite Table Designer</i>	482
SQLite Table Fields	483
Setting SQLite Table Field Properties	485
Setting Other SQLite Table Field Properties	487
SQLite Table Indexes	489
Setting SQLite Table Index Properties	490
SQLite Table Foreign Keys	492
Setting SQLite Table Foreign Key Properties	493
SQLite Table Uniques	495
Setting SQLite Table Unique Properties	496
SQLite Table Checks	499
Setting SQLite Table Check Properties	500
SQLite Table Triggers	501
Setting SQLite Table Trigger Properties	502
SQLite Table Options	503
SQLITE VIEWS	505
<i>SQLite View Designer</i>	509
Working with SQLite View Builder (Available only in Full Version)	510
Editing SQLite View SQL Definition	511
SQLite View Preview	512
SQLite View Explain	513
<i>SQLite View Viewer</i>	514
SQLITE INDEXES	515
<i>SQLite Index Designer</i>	517
Editing SQLite Index General	518
SQLITE TRIGGERS	519
<i>SQLite Trigger Designer</i>	521
Editing SQLite Trigger General	522
Editing SQLite Trigger Definition	523
<b>SQL SERVER DATABASE OBJECT MANAGEMENT</b>	<b>524</b>
SQL SERVER SCHEMAS	525
<i>SQL Server Schema Designer</i>	527
Editing SQL Server Schema General	528
<i>SQL Server Tables</i>	529
SQL Server Table Designer	533
SQL Server Table Fields	534
Setting SQL Server Table Field Properties	536
Setting Other SQL Server Table Field Properties	538

SQL Server Table Indexes	540
Setting SQL Server Table Index Properties	541
SQL Server Table Foreign Keys	543
Setting SQL Server Table Foreign Key Properties	544
SQL Server Table Uniques	546
Setting SQL Server Table Unique Properties	547
SQL Server Table Checks	548
Setting SQL Server Table Check Properties	549
SQL Server Table Triggers	550
Setting SQL Server Table Trigger Properties	551
SQL Server Table Options	553
SQL Server Table Storage	554
<i>SQL Server Views</i>	<i>556</i>
SQL Server View Designer	560
Working with SQL Server View Builder (Available only in Full Version)	561
Editing SQL Server View SQL Definition	562
Setting Advanced SQL Server View Properties	563
SQL Server View Preview	564
SQL Server View Explain	565
SQL Server View Viewer	567
<i>SQL Server Functions/Procedures</i>	<i>569</i>
SQL Server Function Wizard	572
Setting SQL Server Routine Type	573
Setting Parameters for SQL Server Procedure/Function	574
Setting Return Type for SQL Server Function	575
Setting Options for SQL Server Procedure/Function	576
SQL Server Function/Procedure Designer	577
Editing SQL Server Function/Procedure Definition	578
Viewing SQL Server Function/Procedure Result	579
<i>SQL Server Indexes</i>	<i>580</i>
SQL Server Index Designer	583
Editing SQL Server Nonclustered Index Properties	584
Editing SQL Server Nonclustered Index General	585
Editing SQL Server Nonclustered Index Filter	586
Editing Advanced SQL Server Nonclustered Index Properties	587
Editing SQL Server Nonclustered Index Storage	589
Editing SQL Server Clustered Index Properties	590
Editing SQL Server Clustered Index General	591



Editing Advanced SQL Server Clustered Index Properties	592
Editing SQL Server Clustered Index Storage	594
Editing SQL Server XML Index Properties	596
Editing SQL Server XML Index General	597
Editing Advanced SQL Server XML Index Properties	598
Editing SQL Server Spatial Index Properties	599
Editing SQL Server Spatial Index General	600
Editing Advanced SQL Server Spatial Index Properties	602
<i>SQL Server Synonyms</i>	<i>603</i>
SQL Server Synonym Designer	605
Editing SQL Server Synonym General	606
<i>SQL Server Triggers</i>	<i>607</i>
SQL Server Trigger Designer	610
Editing SQL Server Trigger General	611
Setting Advanced SQL Server Trigger Properties	612
Editing SQL Server Trigger Definition	613
SQL SERVER LINKED SERVERS	614
<i>SQL Server Linked Server Designer</i>	<i>616</i>
Editing SQL Server Linked Server General	617
Editing SQL Server Linked Server Security	618
Setting Advanced SQL Server Linked Server Properties	619
SQL SERVER SERVER TRIGGERS	621
<i>SQL Server Server Trigger Designer</i>	<i>623</i>
Editing SQL Server Server Trigger General	624
Setting Advanced SQL Server Server Trigger Properties	625
Editing SQL Server Server Trigger Definition	626
SQL SERVER ASSEMBLIES	627
<i>SQL Server Assembly Designer</i>	<i>629</i>
Editing SQL Server Assembly General	630
SQL SERVER DATABASE TRIGGERS	631
<i>SQL Server Database Trigger Designer</i>	<i>633</i>
Editing SQL Server Database Trigger General	634
Setting Advanced SQL Server Database Trigger Properties	635
Editing SQL Server Database Trigger Definition	636
SQL SERVER PARTITION FUNCTIONS	637
<i>SQL Server Partition Function Designer</i>	<i>639</i>
Editing SQL Server Partition Function General	640
SQL SERVER PARTITION SCHEMES	641

SQL Server Partition Scheme Designer	643
Editing SQL Server Partition Scheme General	644
<b>SQL PREVIEWER</b>	<b>645</b>
<b>TABLE VIEWER</b>	<b>646</b>
DATA VIEW	648
<i>Grid View</i>	649
Using Navigation Bar	650
Limit Record Setting	652
Editing Records	653
Editing Records with Special Handling	654
Editing Records with Foreign Key (Foreign Key Data Selection - Available only in Full Version)	657
Copying/Pasting Data from/into Navicat	659
Sorting and Finding Records	661
Filtering Records (Available only in Full Version)	663
Manipulating Raw Data (Available only for MySQL, PostgreSQL, SQLite and SQL Server)	664
Formatting Table Grid	665
<i>Form View (Available only in Full Version)</i>	668
<i>Text/Blob/BFile View</i>	669
Viewing/Editing Text/BFile field as Memo	670
Viewing/Editing Text/Blob/BFile field as Hexadecimal	671
Viewing/Editing Blob/BFile field as Graphical Image	672
FILTER WIZARD (AVAILABLE ONLY IN FULL VERSION)	673
<i>Adding New Filter Condition</i>	674
<i>Setting Filter Criteria</i>	675
<i>Setting Filter Operator</i>	676
<i>Setting Filter Criteria Values</i>	677
<i>Setting Filter Group</i>	678
<i>Applying Filter Conditions</i>	679
<b>QUERIES</b>	<b>680</b>
QUERY BUILDER (AVAILABLE ONLY IN FULL VERSION)	684
<i>Working with Diagram Area</i>	685
<i>Setting Field Association</i>	686
<i>Setting Output Fields</i>	687
<i>Setting Criteria</i>	688
<i>Setting Grouping Criteria</i>	689
<i>Setting Sorting Criteria</i>	690
<i>Setting Limit Criteria (Available only for MySQL, PostgreSQL and SQLite)</i>	691

QUERY EDITOR	692
<i>Editor View and More Features</i>	694
Code-Completion (Available only in Full Version)	695
SQL Formatting (Available only in Full Version)	697
Code Folding	698
Brace Highlight	699
Find and Replace	700
Copy With Quotes	702
Zoom In/Zoom Out	703
QUERY RESULTS	704
<i>Query Profile and Status (Available only for MySQL)</i>	706
QUERY EXPLAIN	707
<i>Explain Plan for MySQL</i>	708
<i>Explain Plan for Oracle</i>	709
<i>Explain Plan for PostgreSQL</i>	710
<i>Explain Plan for SQLite</i>	711
<i>Explain Plan for SQL Server</i>	712
QUERY PARAMETERS	714
DEBUGGING ORACLE QUERY (AVAILABLE ONLY IN FULL VERSION)	715
<b>DATA MODELING TOOLS (AVAILABLE ONLY IN NAVICAT PREMIUM AND ENTERPRISE VERSION)</b>	<b>716</b>
MODEL DESIGNER	719
<i>Model Tree Palette</i>	720
<i>Diagram Tree Palette</i>	722
<i>Model Diagram Palette</i>	724
Creating Tables	725
Creating Layers	727
Creating Notes and Labels	729
Creating Images	731
Creating Relations	733
Formatting Diagram	735
<i>Properties Palette</i>	737
<i>History Palette</i>	739
<i>Model Preview and Print</i>	740
EXPORT SQL	741
<i>General Settings for Export SQL</i>	742
<i>Advanced Settings for Export SQL</i>	743
SYNCHRONIZE TO DATABASE	744

<i>General Settings for Synchronize to Database</i>	745
<i>Synchronize to Database Result</i>	747
<i>Synchronize to Database Message Log</i>	749
MODEL HINTS AND TIPS	750
<b>DATA MANAGEMENT TOOLS</b>	<b>752</b>
IMPORT WIZARD	754
<i>Setting Import File Format (Step 1)</i>	755
<i>Setting Source File Name (Step 2)</i>	756
Importing ODBC Data (Step 2)	758
Importing MSSQL Data (Step 2)	761
<i>Setting Additional Options for Specific File Type</i>	763
Setting Delimiter (Step 3) - TXT, XML	764
Setting Data Format (Step 4) - TXT, XML, Excel, HTML	767
<i>Setting Target Table (Step 5)</i>	768
<i>Adjusting Field Structures and Mapping Fields (Step 6)</i>	770
<i>Selecting Import Mode (Step 7)</i>	773
<i>Saving and Confirming Import (Step 8)</i>	775
EXPORT WIZARD	776
<i>Setting Export File Format (Step 1)</i>	777
<i>Setting Destination File Name and Encoding (Step 2)</i>	778
<i>Selecting Fields for Export (Step 3)</i>	780
<i>Setting Data Format (Step 4)</i>	781
<i>Saving and Confirming Export (Step 5)</i>	783
DATA TRANSFER (AVAILABLE ONLY IN FULL VERSION)	784
<i>General Settings for Data Transfer</i>	786
<i>Advanced Settings for Same Server Type Data Transfer</i>	787
<i>Advanced Settings for Cross Server Data Transfer (Available only in Navicat Premium)</i>	790
Advanced Settings for Transferring from MySQL to Oracle	791
Advanced Settings for Transferring from MySQL to PostgreSQL	793
Advanced Settings for Transferring from MySQL to SQLite	795
Advanced Settings for Transferring from MySQL to SQL Server	797
Advanced Settings for Transferring from Oracle to MySQL	799
Advanced Settings for Transferring from Oracle to PostgreSQL	801
Advanced Settings for Transferring from Oracle to SQLite	803
Advanced Settings for Transferring from Oracle to SQL Server	805
Advanced Settings for Transferring from PostgreSQL to MySQL	807
Advanced Settings for Transferring from PostgreSQL to Oracle	809
Advanced Settings for Transferring from PostgreSQL to SQLite	811

Advanced Settings for Transferring from PostgreSQL to SQL Server	813
Advanced Settings for Transferring from SQLite to MySQL Database	815
Advanced Settings for Transferring from SQLite to Oracle	817
Advanced Settings for Transferring from SQLite to PostgreSQL	819
Advanced Settings for Transferring from SQLite to SQL Server	821
Advanced Settings for Transferring from SQL Server to MySQL	823
Advanced Settings for Transferring from SQL Server to Oracle	825
Advanced Settings for Transferring from SQL Server to PostgreSQL	827
Advanced Settings for Transferring from SQL Server to SQLite	829
<i>Data Transfer Message Log</i>	831
DATA SYNCHRONIZATION (AVAILABLE ONLY IN FULL VERSION)	832
<i>General Settings for Data Synchronization</i>	835
<i>Advanced Settings for Data Synchronization</i>	836
<i>Data Synchronization Message Log</i>	837
STRUCTURE SYNCHRONIZATION (AVAILABLE ONLY IN FULL VERSION & ONLY FOR MYSQL, ORACLE, POSTGRESQL AND SQL SERVER)	838
<i>General Settings for MySQL Structure Synchronization</i>	840
<i>General Settings for Oracle Structure Synchronization</i>	842
<i>General Settings for PostgreSQL Structure Synchronization</i>	844
<i>General Settings for SQL Server Structure Synchronization</i>	846
<i>Structure Synchronization Result</i>	848
<i>Structure Synchronization Message Log</i>	850
BACKUP/RESTORE (AVAILABLE ONLY IN FULL VERSION & ONLY FOR MYSQL, POSTGRESQL AND SQLITE)	851
<i>Backup</i>	854
General Settings for Backup	855
Object Selection for Backup	856
Advanced Settings for Backup	857
Backup Message Log	858
<i>Restore</i>	859
General Settings for Restore	860
Object Selection for Restore	861
Advanced Settings for Restore	862
Restore Message Log	864
<i>Extract SQL</i>	865
BATCH JOB/SCHEDULE (AVAILABLE ONLY IN FULL VERSION)	866
<i>General Settings for Batch Job/Schedule</i>	870
Setting Report Printing	872
<i>Advanced Settings for Batch Job/Schedule</i>	873

<i>Batch Job/Schedule Message Log</i>	874
<i>Batch Job Converter (Available only in Navicat Premium)</i>	875
Selecting Batch Jobs	876
Setting Convert Options	877
Starting Convert	878
CONSOLE	879
<i>MySQL Console</i>	880
Example of Using MySQL Console	882
<i>Oracle Console</i>	883
Example of Using Oracle Console	885
<i>PostgreSQL Console</i>	886
Example of Using PostgreSQL Console	888
<i>SQLite Console</i>	889
Example of Using SQLite Console	891
<i>SQL Server Console</i>	892
Example of Using SQL Server Console	894
DUMP SQL FILE	895
EXECUTE SQL FILE	896
PRINT DATABASE/SCHEMA/TABLE STRUCTURE (AVAILABLE ONLY IN FULL VERSION)	897
LOG FILES	898
<b>REPORT MANAGEMENT TOOLS (AVAILABLE ONLY IN NAVICAT PREMIUM AND ENTERPRISE VERSION)</b>	<b>899</b>
REPORT DESIGNER	904
<i>Report Data</i>	905
Report Query Wizard	906
Report Query Designer	913
Add Search Criteria	914
Create a Group Sum	918
Concatenate Fields	923
Edit SQL	927
<i>Report Design</i>	929
Report Wizard	930
Create a Simple Report	931
Create a Group-Based Report	935
Report Design Toolbars	939
Standard Component Palette	940
Data Component Palette	942
Advanced Component Palette	944

Report Tree	945
Data Tree	946
Align or Space Toolbar	947
Size Toolbar	949
Nudge Toolbar	950
Draw Toolbar	951
Edit Toolbar	952
Standard Toolbar	954
Format Toolbar	955
Dialogs	957
Print Dialog	958
Page Setup Dialog	960
Groups Dialog	962
Print to File Setup Dialog	963
Data Dialog	964
Grid Options	965
Outline Settings	966
Find Text Settings	967
Report Layout	968
<i>Report Preview</i>	971
Preview and Print	972
Search	973
TUTORIAL ON CREATING DIFFERENT TYPES OF REPORTS (AVAILABLE ONLY FOR MySQL)	974
<i>Simple Report</i>	975
<i>Cover Page</i>	979
<i>Simple Table Listing</i>	983
<i>Invoice</i>	987
<i>Order Summary</i>	999
<i>Detailed Order Summary</i>	1006
<i>Sales Statistics</i>	1013
<i>Crosstab</i>	1022
<i>Mailing Label</i>	1029
<i>Form Letter</i>	1033
<i>Photo Album</i>	1038
<b>SERVER SECURITY MANAGEMENT</b>	<b>1044</b>
MySQL SECURITY MANAGEMENT	1045
<i>Privileges Provided by MySQL</i>	1047
<i>MySQL User Designer</i>	1048



Editing MySQL User General	1049
Setting Advanced MySQL User Properties	1050
Setting MySQL User Server Privileges	1052
Setting MySQL User Privileges	1053
ORACLE SECURITY MANAGEMENT	1054
<i>Privileges Provided by Oracle</i>	1058
<i>Oracle User Designer</i>	1059
Editing Oracle User General	1060
Setting Oracle User Membership	1062
Setting Oracle User Quotas	1063
Setting Oracle User Server Privileges	1064
Setting Oracle User Privileges	1065
<i>Oracle Role Designer</i>	1066
Editing Oracle Role General	1067
Setting Oracle Role Membership	1068
Setting Oracle Role Members	1069
Setting Oracle Role Server Privileges	1070
Setting Oracle Role Privileges	1071
POSTGRESQL SECURITY MANAGEMENT	1072
<i>Privileges Provided by PostgreSQL</i>	1077
<i>Manage Users for PostgreSQL Server 7.3 to 8.0</i>	1078
PostgreSQL User Designer	1079
Editing PostgreSQL User General	1080
Setting PostgreSQL User Membership	1081
Setting PostgreSQL User Privileges	1082
PostgreSQL Group Designer	1083
Editing PostgreSQL Group General	1084
Setting PostgreSQL Group Members	1085
Setting PostgreSQL Group Privileges	1086
<i>Manage Users for PostgreSQL Server 8.1 to 9.1</i>	1087
PostgreSQL Role Designer	1088
Editing PostgreSQL Role General	1089
Setting PostgreSQL Role Membership	1091
Setting PostgreSQL Role Members	1092
Setting PostgreSQL Role Privileges	1093
SQL SERVER SECURITY MANAGEMENT	1094
<i>Privileges Provided by SQL Server</i>	1102
<i>SQL Server Login Designer</i>	1104

Editing SQL Server Login General	1105
Setting SQL Server Login Roles	1108
Setting SQL Server Login User Mapping	1109
Setting SQL Server Login Server Permissions	1110
Setting SQL Server Login Endpoint Permissions	1111
Setting SQL Server Login Login Permissions	1112
<i>SQL Server Server Role Designer</i>	<i>1113</i>
Editing SQL Server Server Role General	1114
<i>SQL Server Database User Designer</i>	<i>1115</i>
Editing SQL Server Database User General	1116
Setting SQL Server Database User Roles	1118
Setting SQL Server Database User Owned Schemas	1119
Setting SQL Server Database User Database Permissions	1120
Setting SQL Server Database User Object Permissions	1121
<i>SQL Server Database Role Designer</i>	<i>1122</i>
Editing SQL Server Database Role General	1123
Setting SQL Server Database Role Owned Schemas	1124
Setting SQL Server Database Role Object Permissions	1125
<i>SQL Server Application Role Designer</i>	<i>1126</i>
Editing SQL Server Application Role General	1127
Setting SQL Server Application Role Owned Schemas	1128
Setting SQL Server Application Role Object Permissions	1129
PRIVILEGE MANAGER	1130
<b>DATABASE MAINTENANCE TASKS</b>	<b>1131</b>
MySQL MAINTENANCE TASKS	1132
<i>Analyze MySQL Tables</i>	<i>1133</i>
<i>Check MySQL Tables</i>	<i>1134</i>
<i>Optimize MySQL Tables</i>	<i>1135</i>
<i>Repair MySQL Tables</i>	<i>1136</i>
<i>Flush MySQL</i>	<i>1137</i>
ORACLE MAINTENANCE TASKS	1138
<i>Oracle Table Maintenance Tasks</i>	<i>1139</i>
Enable / Disable Table Lock	1140
Enable / Disable Row Movement	1141
Shrink Space	1142
Move	1143
Collect Statistics	1144
Validate Structure	1145

<i>Oracle View Maintenance Tasks</i>	1146
<i>Oracle Function/Procedure Maintenance Tasks</i>	1147
<i>Oracle Index Maintenance Tasks</i>	1148
<i>Oracle Java Maintenance Tasks</i>	1149
<i>Oracle Materialized View Maintenance Tasks</i>	1150
<i>Oracle Materialized View Log Maintenance Tasks</i>	1151
<i>Oracle Package Maintenance Tasks</i>	1152
<i>Oracle Trigger Maintenance Tasks</i>	1153
<i>Oracle Type Maintenance Tasks</i>	1154
<i>Oracle XML Schema Maintenance Tasks</i>	1155
<i>Oracle Tablespace Maintenance Tasks</i>	1156
POSTGRESQL MAINTENANCE TASKS	1157
<i>Analyze PostgreSQL Database and Tables</i>	1158
<i>Vacuum PostgreSQL Database and Tables</i>	1159
<i>Reindex PostgreSQL Database and Tables</i>	1160
SQLITE MAINTENANCE TASKS	1161
<i>SQLite Database and Table Maintenance Tasks</i>	1162
Analyze SQLite Database and Tables	1163
Vacuum SQLite Database	1164
Reindex SQLite Database and Tables	1165
View Master Table	1166
<i>SQLite Index Maintenance Tasks</i>	1167
SQL SERVER MAINTENANCE TASKS	1168
<i>SQL Azure Firewall Rules (Available only for SQL Azure)</i>	1169
<i>SQL Server Assembly Maintenance Tasks</i>	1170
<i>SQL Server Index Maintenance Tasks</i>	1171
<i>SQL Server Trigger Maintenance Tasks</i>	1172
SERVER MONITOR (AVAILABLE ONLY IN FULL VERSION & ONLY FOR MySQL, ORACLE, POSTGRESQL AND SQL SERVER)	1173
<i>Process List</i>	1174
<i>Variables (Available only for MySQL, Oracle and PostgreSQL)</i>	1175
<i>Status (Available only for MySQL, Oracle and PostgreSQL)</i>	1176
<b>OPTIONS</b>	<b>1177</b>
GENERAL OPTIONS	1178
<i>Docking Options</i>	1179
<i>Code Insight Options (Available only in Full Version)</i>	1180
<i>Auto Save Options</i>	1181
APPEARANCE OPTIONS	1182

<i>Color Options</i>	1183
<i>Main Window Options</i>	1184
<i>Editor Options</i>	1185
<i>Data/Grid Options</i>	1186
Display Format Options	1188
MODEL OPTIONS (AVAILABLE ONLY IN FULL VERSION)	1193
MISCELLANEOUS OPTIONS	1194
<i>File Association Options (Available only in Full Version)</i>	1195
<i>OCI Options (Available only for Oracle)</i>	1196
<b>USEFUL TOOLS</b>	<b>1197</b>
ER DIAGRAM (AVAILABLE ONLY IN FULL VERSION)	1198
VIRTUAL GROUPING (AVAILABLE ONLY IN FULL VERSION)	1201
CONNECTION COLORINGS	1203
FAVORITES (AVAILABLE ONLY IN FULL VERSION)	1204
FIND IN DATABASE (AVAILABLE ONLY IN FULL VERSION)	1205
SEARCH FILTER	1206
<b>NAVICAT COMMANDS (AVAILABLE ONLY IN FULL VERSION)</b>	<b>1207</b>
<b>NAVICAT SUPPORT INFORMATION</b>	<b>1216</b>

## Welcome to Navicat!

### **Navicat Premium**

Navicat Premium is a multi-connections Database Administration tool allowing you to connect to MySQL, Oracle, PostgreSQL, SQLite and SQL Server databases simultaneously within a single application, making database administration to multiple kinds of database so easy. Navicat Premium combines the functions of other Navicat members. With connections established to different database types, Navicat Premium supports data transfer between MySQL, Oracle, PostgreSQL, SQLite and SQL Server. It supports most of the features in MySQL, Oracle, PostgreSQL, SQLite and SQL Server.

Navicat Premium enables you to easily and quickly transfer data across various database systems, or to a plain text file with designated SQL format and encoding. This can simplified the process for migrating data from one server to another server type. Batch job for different kind of databases can also be scheduled and run at a specific time.

### **Navicat for MySQL**

Navicat for MySQL is a high performance tool for MySQL database administration and development. It works with any MySQL version from 3.21 or above and supports most of the latest MySQL features including Trigger, Procedure, Function, Event, View, Manage User, and so on.

### **Navicat for Oracle**

Navicat for Oracle is a powerful Oracle Database Server administration and development tool. It works with any Oracle version from 8.1 to the newest one and supports all Oracle objects including Directory, Tablespace, Synonym, Materialized View, Trigger, Sequence, Type and so on.

### **Navicat for PostgreSQL**

Navicat for PostgreSQL is a powerful PostgreSQL Database Server administration and development tool. It works with PostgreSQL version 7.3 or above and supports most of the PostgreSQL features including Trigger, Function, View, Manage User, and so on.

## Navicat for SQLite

Navicat for SQLite is a powerful SQLite Database Server administration and development tool. It works with SQLite version 2 and 3 and supports most of the SQLite features including Trigger, Index, View, and so on.

## Navicat for SQL Server

Navicat for SQL Server is a high performance tool for Microsoft SQL Server database development and administration. It works with SQL Server 2000, 2005, 2008R2 and SQL Azure, and supports most of the SQL Server features including Trigger, Index, View, and so on.

**Hint:** If you need to manage more than one kind of servers, then you should choose Navicat Premium. Otherwise, you can choose Navicat for MySQL, Navicat for Oracle, Navicat for PostgreSQL, Navicat for SQLite or Navicat for SQL Server according to your server type.

## Navicat Feature

Features in Navicat are sophisticated enough to provide professional developers for all their specific needs, yet easy to learn for users who are new to database server. With its well-designed Graphical User Interface(GUI), Navicat lets you quickly and easily create, organize, access and share information in a secure and easy way.

Navicat is available on three platforms - Microsoft Windows, Mac OS X and Linux. It can connect users to local/remote server, providing several utility tools such as Data Modeling, Data Transfer, Data/Structure Synchronization, Import/Export, Backup/Restore, Report Builder and Schedule to facilitate the process for data maintenance. See the [Feature Matrix](http://www.navicat.com) for details or visit our web-site: <http://www.navicat.com>

## System Requirements

### System Requirements for Windows

- Microsoft Windows XP SP2, Vista, Server 2003, Server 2008, Windows 7
- Pentium II processor or above
- Hard-disk space vary depending on installation. Full installation requires 65 MB of available hard-disk space.

### System Requirements for Mac OS X

- Mac OS X 10.5 Leopard, 10.6 Snow Leopard or 10.7 Lion
- Intel CPU

### System Requirements for Linux

- Compatible with i386 PC
- Supports 32-bit and 64-bit Linux platform
- Supports Linux kernel version 2.2 or higher
- Supports Glibc 2.4 or above
- Supports GNOME and KDE

**Note:** You need to install all 32-bit libraries before working on 64-bit Linux.



## Feature Matrix

Navicat provides the following versions for managing your servers.

- [Navicat Premium](#)
- [Navicat for MySQL](#)
- [Navicat for Oracle](#)
- [Navicat for PostgreSQL](#)
- [Navicat for SQLite](#)
- [Navicat for SQL Server](#)

## Navicat Premium

**Note:** ESS stands for Navicat Essentials.

Platforms	Win	Win ESS	Mac	Mac ESS	Linux	Linux ESS
<b>MySQL Objects Management</b>						
Support MySQL Servers 3.21 or above	+	+	+	+	+	+
Support of Drizzle, OurDelta, Percona Server and MariaDB	+	+	+	+	+	+
Create / Drop Databases	+	+	+	+	+	+
Support all MySQL objects: tables, views, procedures/functions and events	+	+	+	+	+	+
Support all subobjects: fields, indexes, foreign keys and triggers	+	+	+	+	+	+
Support of partitioning	+	+	+	+	+	+
Support of character set and unicode	+	+	+	+	+	+
<b>Oracle Objects Management</b>						
Support Oracle Servers 8.1 or above (Mac OS X Edition starts from Oracle 8.1.7)	+	+	+	+	+	+
Create/drop schemas	+	+	+	+	+	+
Support all Oracle objects: tables (Normal, External and Index Organized), views, procedures/functions	+	+	+	+	+	+
Support all subobjects: fields, indexes, foreign keys, uniques, checks and triggers	+	+	+	+	+	+
Managing directories, tablespaces, public database links and public synonyms	+	+	+	+	+	+
Managing database links, indexes, java, materialized views, materialized view logs, packages, sequences, synonyms, triggers, types, XML Schema and Recycle Bin	+	+	+	+	+	+
Support of physical attributes	+	+	+	+	+	+
Support of character set and unicode	+	+	+	+	+	+
<b>PostgreSQL Objects Management</b>						
Support PostgreSQL Servers 7.3 or above	+	+	+	+	+	+
Create /drop databases and schemas	+	+	+	+	+	+

Support all PostgreSQL objects: tables, views and functions	+	+	+	+	+	+
Support all subobjects: fields, indexes, foreign keys, uniques, checks, rules and triggers	+	+	+	+	+	+
Managing tablespace	+	+	+	+	+	+
Managing cast and languages	+	+	+	+	+	+
Managing aggregates, conversions, domains, trigger functions, operators and classes, sequences and types	+	+	+	+	+	+
Support of character set and unicode	+	+	+	+	+	+
<b>SQLite Objects Management</b>						
Support SQLite 2 and 3	+	+	+	+	+	+
Attach/detach databases	+	+	+	+	+	+
Support all SQLite objects: tables and views	+	+	+	+	+	+
Support all subobjects: triggers and indexes	+	+	+	+	+	+
Support to view master table	+	+	+	+	+	+
Support of character set and unicode	+	+	+	+	+	+
<b>SQL Server Objects Management</b>						
Support SQL Server 2000 or above	+	+	+	+	-	-
Support SQL Azure	+	+	+	+	-	-
Create /drop databases and schemas	+	+	+	+	-	-
Support all SQL Server objects: tables, views and functions	+	+	+	+	-	-
Support all subobjects: fields, indexes, foreign keys, uniques, checks and triggers	+	+	+	+	-	-
Managing linked servers, server triggers	+	+	+	+	-	-
Managing assemblies, partition functions, partition schemes and database triggers	+	+	+	+	-	-
Managing indexes, synonyms and triggers	+	+	+	+	-	-
Support of character set and unicode	+	+	+	+	-	-
<b>Connection Management and Navigation</b>						
Multiple local/remote server connections	+	+	+	+	+	+
Support of Oracle Basic Connection	+	+	+	+	+	+
Support of Oracle TNS Connection	+	+	+	+	-	-
Support to connect a new or an existing SQLite database	+	+	+	+	+	+
Support encrypted SQLite file (wxSQLite3)	+	+	+	+	+	+

Connection to MySQL/Oracle/PostgreSQL/SQL Server through SSH	+	+	+	+	+	+
Connection to MySQL/PostgreSQL/SQLite through HTTP	+	+	+	+	+	+
Support SSL secure connection for MySQL and PostgreSQL 8.4 above	+	+	+	+	+	+
Keep connection alive	+	+	+	+	+	+
Export/Import connections settings	+	+	+	+	+	+
Customize selected MySQL/PostgreSQL databases to work with	+	+	+	+	+	+
<b>Feature Rich Oracle PL/SQL Code Debugger</b>						
Support on showing interactive and real-time debugging information	+	-	+	-	+	-
Ability to set simple and complex breakpoints	+	-	+	-	+	-
Stepping through a function, i.e. Step Into, Step Over and Step Out	+	-	+	-	+	-
Inspecting variables and expressions	+	-	+	-	+	-
Stack traces	+	-	+	-	+	-
<b>Data Viewer and Editor</b>						
UUID/GUID generator for PostgreSQL/SQL Server	+	+	+	+	+	+
Grid view	+	+	+	+	+	+
Form view	+	-	+	-	+	-
Text, Hex and BLOB viewer/editor	+	+	+	+	+	+
Oracle BFile viewer/editor	+	+	+	+	+	+
Copying and pasting selected records	+	+	+	+	+	+
Customize display formats	+	+	+	+	+	+
Incremental search, sorting and formatting table grid	+	+	+	+	+	+
Ability to show/hide Oracle and SQLite tables ROWID	+	+	+	+	+	+
Support of raw mode edit: apply MySQL/PostgreSQL/SQLite/SQL Server built-in function	+	+	+	+	+	+
Support of foreign key data selection: locate the available values from the reference table	+	-	+	-	+	-
Filtering Records	+	-	+	-	+	-
Ability to set number of records showing on each page	+	+	+	+	+	+
<b>SQL Processing</b>						
Visual query/view builder to facilitate the process of building complicated queries	+	-	+	-	+	-

Ability to create parameter queries	+	+	+	+	+	+
Code completion for SQL editors	+	-	+	-	+	-
Code outline	+	-	+	-	+	-
SQL Code Beautifier/Minifier	+	+	+	+	+	+
Multiple SQL editors with syntax highlighted feature	+	+	+	+	+	+
Support code folding, highlighted matches words and brackets	+	+	+	+	+	+
Copy SQL with quote	+	+	+	+	+	+
Word Wrap	+	+	+	+	+	+
Zoom In/Out for SQL editors	+	+	+	+	+	+
Enhanced find and replace options in SQL editors	+	+	+	+	+	+
Ability to preview SQL before execution	+	+	+	+	+	+
Support to return multi-resultsets	+	+	-	-	+	+
SQL console	+	+	+	+	-	-
<b>Data Modeling Tools</b>						
Reverse engineering	+	-	+	-	+	-
Support of several diagram notations like Crow's Foot, IDEF1x and UML	+	-	+	-	+	-
Ability to create/edit/drop database objects, visually set relationships between them	+	-	+	-	+	-
Support of auto layout	+	-	+	-	+	-
Support numerous visual customization options like adding vertices, layers, images and notes/labels	+	-	+	-	+	-
Zoom In/Out for ER diagram	+	-	+	-	+	-
Comparing and synchronizing database/model	+	-	+	-	+	-
Generating complex SQL/DDDL	+	-	+	-	+	-
Multi-page print preview	+	-	-	-	+	-
Ability to print the database diagram to PDF file	+	-	+	-	+	-
Ability to save the database diagram as a graphic file, i.e. PNG, SVG	+	-	+	-	+	-
<b>Diversified Import and Export Competence</b>						
Importing data from ODBC: MSSQL, Oracle etc	+	-	+	-	-	-
Importing data from MS Excel	+	-	+	-	-	-
Importing data from MS Access	+	-	-	-	-	-
Importing data from plain text file formats : TXT, CSV, XML etc	+	+	+	+	+	+

Importing data from other file formats : DBF etc	+	-	+	-	+	-
Exporting data to MS Excel	+	-	+	-	+	-
Exporting data to MS Access	+	-	-	-	-	-
Exporting data to HTML	+	-	-	-	+	-
Exporting data to plain text file formats: TXT, CSV, XML etc	+	+	+	+	+	+
Exporting data to most popular formats: DBF etc	+	-	+	-	+	-
<b>Other Data Manipulation Tools</b>						
Data transfer (Same or cross server type)	+	-	+	-	+	-
Transferring data to SQL script as INSERT statement	+	-	+	-	+	-
Transferring data to a SQL file with designated SQL format and encoding	+	-	+	-	+	-
Data synchronization	+	-	+	-	+	-
Structure synchronization	+	-	+	-	+	-
<b>Backup/Restore for MySQL/PostgreSQL/SQLite</b>						
Backing up/restoring database	+	-	+	-	+	-
Backing up database with chosen database objects	+	-	+	-	+	-
Restoring database with chosen database objects	+	-	-	-	+	-
Converting backup file into SQL script	+	-	+	-	+	-
Compressing/Decompressing Backup files	+	-	-	-	+	-
<b>Schedule for Batch Job</b>						
Ability to set schedule on backup, query execution, import/export, data transfer and data synchronization	+	-	+	-	+	-
Ability to run profiles from different servers in a single schedule job	+	-	+	-	+	-
Schedule report printing	+	-	-	-	-	-
Schedule to print report to files: PDF, Excel, HTML etc	+	-	-	-	-	-
Sending notification e-mail for schedule task	+	-	+	-	+	-
Ability to attach Export Wizard/Report result files in Batch Job notification email	+	-	+	-	+	-
Notification email Support SSL/TLS	+	-	-	-	+	-
<b>Server Security and Maintenance Services</b>						
Visual user management	+	+	+	+	+	+
Visual privilege manager	+	+	+	+	+	+
Ability to duplicate user, role, etc	+	+	+	+	+	+
Maintenance database/schema objects	+	+	+	+	+	+

Server Monitor to view and change system variables and server status information	+	-	+	-	+	-
<b>Report Management Tools</b>						
Report Builder	+	-	-	-	-	-
Report Viewer	+	-	-	-	-	-
<b>Other Useful Features</b>						
ER Diagram view	+	-	+	-	+	-
Ability to customize connection colorings	+	+	+	+	+	+
Support of database wide search	+	-	+	-	+	-
Objects search filter	+	+	+	+	+	+
Backing up database/schema/table to SQL script	+	+	+	+	+	+
Executing SQL script	+	+	+	+	+	+
Duplicate/empty/truncate tables	+	+	+	+	+	+
Create table/view shortcut on desktop	+	+	+	+	-	-
Ability to create Favorites list	+	-	+	-	+	-
Ability to set auto-save for Query, Function, etc	+	+	+	+	+	+
Print database/schema/table structure	+	-	+	-	+	-
Support of docking/tab windows	+	+	+	+	-	-
Support of virtual grouping: provide logical organization over connections and objects	+	-	+	-	+	-
Ability to run profile from command line	+	-	+	-	+	-
Log files: keep track on the actions have been performed in Navicat	+	+	+	+	+	+
Ability to copy Connection Information into Clipboard	+	+	+	+	+	+



## Navicat for MySQL

**Note:** ESS stands for Navicat Essentials.

Platforms	Win ENT	Win STD	Win ESS	Mac ENT	Mac STD	Mac ESS	Linux ENT	Linux STD	Linux ESS
<b>MySQL Objects Management</b>									
Support MySQL Servers 3.21 or above	+	+	+	+	+	+	+	+	+
Support of Drizzle, OurDelta, Percona Server and MariaDB	+	+	+	+	+	+	+	+	+
Create / Drop Databases	+	+	+	+	+	+	+	+	+
Support all MySQL objects: tables, views, stored procedures/functions and events	+	+	+	+	+	+	+	+	+
Support all subobjects: fields, indexes, foreign keys and triggers	+	+	+	+	+	+	+	+	+
Support of partitioning	+	+	+	+	+	+	+	+	+
Support of character set and unicode	+	+	+	+	+	+	+	+	+
<b>Connection Management and Navigation</b>									
Multiple local/remote MySQL server connections	+	+	+	+	+	+	+	+	+
Connection to MySQL Server through SSH	+	+	+	+	+	+	+	+	+
Connection to MySQL Server through HTTP	+	+	+	+	+	+	+	+	+
Support SSL secure connection	+	+	+	+	+	+	+	+	+
Keep connection alive	+	+	+	+	+	+	+	+	+
Export/Import connections settings	+	+	+	+	+	+	+	+	+
Customize selected databases to work with	+	+	+	+	+	+	+	+	+
<b>Data Viewer and Editor</b>									
Grid view	+	+	+	+	+	+	+	+	+
Form view	+	+	-	+	+	-	+	+	-
Text, Hex and BLOB viewer/editor	+	+	+	+	+	+	+	+	+
Copying and pasting selected records	+	+	+	+	+	+	+	+	+
Customize display formats	+	+	+	+	+	+	+	+	+

Incremental search, sorting and formatting table grid	+	+	+	+	+	+	+	+	+
Support of raw mode edit: apply MySQL built-in function	+	+	+	+	+	+	+	+	+
Support of foreign key data selection: locate the available values from the reference table	+	+	-	+	+	-	+	+	-
Filtering Records	+	+	-	+	+	-	+	+	-
Ability to set number of records showing on each page	+	+	+	+	+	+	+	+	+
<b>SQL Processing</b>									
Visual query/view builder to facilitate the process of building complicated queries	+	+	-	+	+	-	+	+	-
Ability to create parameter queries	+	+	+	+	+	+	+	+	+
Code completion for SQL editors	+	+	-	+	+	-	+	+	-
SQL Code Beautifier/Minifier	+	+	+	+	+	+	+	+	+
Multiple SQL editors with syntax highlighted feature	+	+	+	+	+	+	+	+	+
Support code folding, highlighted matches words and brackets	+	+	+	+	+	+	+	+	+
Copy SQL with quote	+	+	+	+	+	+	+	+	+
Word Wrap	+	+	+	+	+	+	+	+	+
Zoom In/Out for SQL editors	+	+	+	+	+	+	+	+	+
Enhanced find and replace options in SQL editors	+	+	+	+	+	+	+	+	+
Ability to preview SQL before execution	+	+	+	+	+	+	+	+	+
Support to return multi-resultsets	+	+	+	-	-	-	+	+	+
SQL console	+	+	+	+	+	+	-	-	-
<b>Data Modeling Tools</b>									
Reverse engineering	+	-	-	+	-	-	+	-	-
Support of several diagram notations like Crow's Foot, IDEF1x and UML	+	-	-	+	-	-	+	-	-
Ability to create/edit/drop database objects, visually set relationships between them	+	-	-	+	-	-	+	-	-

Support of auto layout	+	-	-	+	-	-	+	-	-
Support numerous visual customization options like adding vertices, layers, images and notes/labels	+	-	-	+	-	-	+	-	-
Zoom In/Out for ER diagram	+	-	-	+	-	-	+	-	-
Comparing and synchronizing database/model	+	-	-	+	-	-	+	-	-
Generating complex SQL/DDDL	+	-	-	+	-	-	+	-	-
Multi-page print preview	+	-	-	-	-	-	+	-	-
Ability to print the database diagram to PDF file	+	-	-	+	-	-	+	-	-
Ability to save the database diagram as a graphic file, i.e. PNG, SVG	+	-	-	+	-	-	+	-	-

## **Diversified Import and Export Competence**

Importing data from ODBC: MSSQL, Oracle etc	+	+	-	+	+	-	-	-	-
Importing data from MS Excel	+	+	-	+	+	-	-	-	-
Importing data from MS Access	+	+	-	-	-	-	-	-	-
Importing data from plain text file formats : TXT, CSV, XML etc	+	+	+	+	+	+	+	+	+
Importing data from other file formats : DBF etc	+	+	-	+	+	-	+	+	-
Exporting data to MS Excel	+	+	-	+	+	-	+	+	-
Exporting data to MS Access	+	+	-	-	-	-	-	-	-
Exporting data to HTML	+	+	-	-	-	-	+	+	-
Exporting data to plain text file formats: TXT, CSV, XML etc	+	+	+	+	+	+	+	+	+
Exporting data to most popular formats: DBF etc	+	+	-	+	+	-	+	+	-

## **Other Data Manipulation Tools**

Data transfer	+	+	-	+	+	-	+	+	-
Transferring data to SQL script as INSERT statement	+	+	-	+	+	-	+	+	-
Transferring data to a SQL file with designated SQL format and encoding	+	+	-	+	+	-	+	+	-
Data synchronization	+	+	-	+	+	-	+	+	-

Structure synchronization	+	+	-	+	+	-	+	+	-
<b>Backup/Restore</b>									
Backing up/restoring database	+	+	-	+	+	-	+	+	-
Backing up database with chosen database objects	+	+	-	+	+	-	+	+	-
Restoring database with chosen database objects	+	+	-	-	-	-	+	+	-
Converting backup file into SQL script	+	+	-	+	+	-	+	+	-
Compressing/Decompressing Backup files	+	+	-	-	-	-	+	+	-
<b>Schedule for Batch Job</b>									
Ability to set schedule on backup, query execution, import/export, data transfer and data synchronization	+	+	-	+	+	-	+	+	-
Schedule report printing	+	-	-	-	-	-	-	-	-
Schedule to print report to files: PDF, Excel, HTML etc	+	-	-	-	-	-	-	-	-
Sending notification e-mail for schedule task	+	+	-	+	+	-	+	+	-
Ability to attach Export Wizard/Report result files in Batch Job notification email	+	+	-	+	+	-	+	+	-
Notification email Support SSL/TLS	+	+	-	-	-	-	+	+	-
<b>Server Security and Maintenance Services</b>									
Visual user management	+	+	+	+	+	+	+	+	+
Visual privilege manager	+	+	+	+	+	+	+	+	+
Ability to duplicate user	+	+	+	+	+	+	+	+	+
Maintenance database objects	+	+	+	+	+	+	+	+	+
Server Monitor to view and change system variables, to view server status information	+	+	-	+	+	-	+	+	-
<b>Report Management Tools</b>									
Report Builder	+	-	-	-	-	-	-	-	-
Report Viewer	+	-	-	-	-	-	-	-	-
<b>Other Useful Features</b>									
ER Diagram view	+	+	-	+	+	-	+	+	-

Ability to customize connection colorings	+	+	+	+	+	+	+	+	+
Support of database wide search	+	+	-	+	+	-	+	+	-
Objects search filter	+	+	+	+	+	+	+	+	+
Backing up database/table to SQL script	+	+	+	+	+	+	+	+	+
Executing SQL script	+	+	+	+	+	+	+	+	+
Duplicate/empty/truncate tables	+	+	+	+	+	+	+	+	+
Create table/view shortcut on desktop	+	+	+	+	+	+	-	-	-
Ability to create Favorites list	+	+	-	+	+	-	+	+	-
Ability to set auto-save for Query, Function, etc	+	+	+	+	+	+	+	+	+
Print database/table Structure	+	+	-	+	+	-	+	+	-
Support of docking/tab windows	+	+	+	+	+	+	-	-	-
Support of virtual grouping: provide logical organization over connections and objects	+	+	-	+	+	-	+	+	-
Ability to run profile from command line	+	+	-	+	+	-	+	+	-
Log files: keep track on the actions have been performed in Navicat	+	+	+	+	+	+	+	+	+
Ability to copy Connection Information into Clipboard	+	+	+	+	+	+	+	+	+

## Navicat for Oracle

**Note:** ESS stands for Navicat Essentials.

Platforms	Win ENT	Win STD	Win ESS	Mac ENT	Mac STD	Mac ESS	Linux ENT	Linux STD	Linux ESS
<b>Oracle Objects Management</b>									
Support Oracle Servers 8.1 or above (Mac OS X Edition starts from Oracle 8.1.7)	+	+	+	+	+	+	+	+	+
Create/drop schemas	+	+	+	+	+	+	+	+	+
Support all Oracle objects: tables (Normal, External and Index Organized), views, stored procedures/functions	+	+	+	+	+	+	+	+	+
Support all subobjects: fields, indexes, foreign keys, uniques, checks and triggers	+	+	+	+	+	+	+	+	+
Managing directories, tablespaces, public database links and public synonyms	+	+	+	+	+	+	+	+	+
Managing database links, indexes, java, materialized views, materialized view logs, packages, sequences, synonyms, triggers, types, XML Schema and Recycle Bin.	+	+	+	+	+	+	+	+	+
Support of physical attributes	+	+	+	+	+	+	+	+	+
Support of character set and unicode	+	+	+	+	+	+	+	+	+
<b>Connection Management and Navigation</b>									
Multiple local/remote Oracle server connections	+	+	+	+	+	+	+	+	+
Support of Basic Connection	+	+	+	+	+	+	+	+	+
Support of TNS Connection	+	+	+	+	+	+	-	-	-
Connection to Oracle Server through SSH	+	+	+	+	+	+	+	+	+
Keep connection alive	+	+	+	+	+	+	+	+	+
Export/Import connections settings	+	+	+	+	+	+	+	+	+

Feature Rich Oracle PL/SQL Code Debugger									
Support on showing interactive and real-time debugging information	+	+	-	+	+	-	+	+	-
Ability to set simple and complex breakpoints	+	+	-	+	+	-	+	+	-
Stepping through a function, i.e. Step Into, Step Over and Step Out	+	+	-	+	+	-	+	+	-
Inspecting variables and expressions	+	+	-	+	+	-	+	+	-
Stack traces	+	+	-	+	+	-	+	+	-
Data Viewer and Editor									
Grid view	+	+	+	+	+	+	+	+	+
Form view	+	+	-	+	+	-	+	+	-
TEXT, Hex and BLOB/BFile viewer/editor	+	+	+	+	+	+	+	+	+
Copying and pasting selected records	+	+	+	+	+	+	+	+	+
Customize display formats	+	+	+	+	+	+	+	+	+
Incremental search, sorting and formatting table grid	+	+	+	+	+	+	+	+	+
Ability to show/hide table ROWID	+	+	+	+	+	+	+	+	+
Support of foreign key data selection: locate the available values from the reference table	+	+	-	+	+	-	+	+	-
Filtering Records	+	+	-	+	+	-	+	+	-
Ability to set number of records showing on each page	+	+	+	+	+	+	+	+	+
SQL Processing									
Visual query/view builder to facilitate the process of building complicated queries	+	+	-	+	+	-	+	+	-
Ability to create parameter queries	+	+	+	+	+	+	+	+	+
Code completion for SQL editors	+	+	-	+	+	-	+	+	-
Code outline	+	+	-	+	+	-	+	+	-
SQL Code Beautifier/Minifier	+	+	+	+	+	+	+	+	+
Multiple SQL editors with syntax highlighted feature	+	+	+	+	+	+	+	+	+

Support code folding, highlighted matches words and brackets	+	+	+	+	+	+	+	+	+
Copy SQL with quote	+	+	+	+	+	+	+	+	+
Word Wrap	+	+	+	+	+	+	+	+	+
Zoom In/Out for SQL editors	+	+	+	+	+	+	+	+	+
Enhanced find and replace options in SQL editors	+	+	+	+	+	+	+	+	+
Ability to preview SQL before execution	+	+	+	+	+	+	+	+	+
Support to return multi-resultsets	+	+	+	-	-	-	+	+	+
SQL*Plus	+	+	+	+	+	+	-	-	-
<b>Data Modeling Tools</b>									
Reverse engineering	+	-	-	+	-	-	+	-	-
Support of several diagram notations like Crow's Foot, IDEF1x and UML	+	-	-	+	-	-	+	-	-
Ability to create/edit/drop database objects, visually set relationships between them	+	-	-	+	-	-	+	-	-
Support of auto layout	+	-	-	+	-	-	+	-	-
Support numerous visual customization options like adding vertices, layers, images and notes/labels	+	-	-	+	-	-	+	-	-
Zoom In/Out for ER diagram	+	-	-	+	-	-	+	-	-
Comparing and synchronizing database/model	+	-	-	+	-	-	+	-	-
Generating complex SQL/DDDL	+	-	-	+	-	-	+	-	-
Multi-page print preview	+	-	-	-	-	-	+	-	-
Ability to print the database diagram to PDF file	+	-	-	+	-	-	+	-	-
Ability to save the database diagram as a graphic file, i.e. PNG, SVG	+	-	-	+	-	-	+	-	-
<b>Diversified Import and Export Competence</b>									
Importing data from ODBC: MSSQL etc	+	+	-	+	+	-	-	-	-
Importing data from MS Excel	+	+	-	+	+	-	-	-	-
Importing data from MS Access	+	+	-	-	-	-	-	-	-
Importing data from plain text file formats : TXT, CSV, XML etc	+	+	+	+	+	+	+	+	+



Importing data from other file formats : DBF etc	+	+	-	+	+	-	+	+	-
Exporting data to MS Excel	+	+	-	+	+	-	+	+	-
Exporting data to MS Access	+	+	-	-	-	-	-	-	-
Exporting data to HTML	+	+	-	-	-	-	+	+	-
Exporting data to plain text file formats : TXT, CSV, XML etc	+	+	+	+	+	+	+	+	+
Exporting data to most popular formats: DBF etc	+	+	-	+	+	-	+	+	-
<b>Other Data Manipulation Tools</b>									
Data transfer	+	+	-	+	+	-	+	+	-
Transferring data to SQL script as INSERT statement	+	+	-	+	+	-	+	+	-
Transferring data to a SQL file with designated SQL format and encoding	+	+	-	+	+	-	+	+	-
Data synchronization	+	+	-	+	+	-	+	+	-
Structure synchronization	+	+	-	+	+	-	+	+	-
<b>Schedule for Batch Job</b>									
Ability to set schedule on query execution, import/export, data transfer and data synchronization	+	+	-	+	+	-	+	+	-
Schedule report printing	+	-	-	-	-	-	-	-	-
Schedule to print report to files: PDF, Excel, HTML etc	+	-	-	-	-	-	-	-	-
Sending notification e-mail for schedule task	+	+	-	+	+	-	+	+	-
Ability to attach Export Wizard/Report result files in Batch Job notification email	+	+	-	+	+	-	+	+	-
Notification email Support SSL/TLS	+	+	-	-	-	-	+	+	-
<b>Server Security and Maintenance Services</b>									
Visual user management	+	+	+	+	+	+	+	+	+
Visual privilege manager	+	+	+	+	+	+	+	+	+
Ability to duplicate user, role	+	+	+	+	+	+	+	+	+

Maintaining tables like Table Lock, Row Movement, Shrink Space, Table Move, Validate Table Structure and Collect Table Statistics	+	+	+	+	+	+	+	+	+
Server Monitor to view and change system variables, to view server status information	+	+	-	+	+	-	+	+	-
<b>Report Management Tools</b>									
Report Builder	+	-	-	-	-	-	-	-	-
Report Viewer	+	-	-	-	-	-	-	-	-
<b>Other Useful Features</b>									
ER Diagram view	+	+	-	+	+	-	+	+	-
Ability to customize connection colorings	+	+	+	+	+	+	+	+	+
Support of database wide search	+	+	-	+	+	-	+	+	-
Objects search filter	+	+	+	+	+	+	+	+	+
Backing up schema/table to SQL script	+	+	+	+	+	+	+	+	+
Executing SQL script	+	+	+	+	+	+	+	+	+
Duplicate/empty/truncate tables	+	+	+	+	+	+	+	+	+
Create table/view shortcut on desktop	+	+	+	+	+	+	-	-	-
Ability to create Favorites list	+	+	-	+	+	-	+	+	-
Ability to set auto-save for Query, Function, etc	+	+	+	+	+	+	+	+	+
Print Schema Structure	+	+	-	+	+	-	+	+	-
Support of docking/tab windows	+	+	+	+	+	+	-	-	-
Support of virtual grouping: provide logical organization over connections and objects	+	+	-	+	+	-	+	+	-
Ability to run profile from command line	+	+	-	+	+	-	+	+	-
Log files: keep track on the actions have been performed in Navicat	+	+	+	+	+	+	+	+	+
Ability to copy Connection Information into Clipboard	+	+	+	+	+	+	+	+	+

## Navicat for PostgreSQL

**Note:** ESS stands for Navicat Essentials.

Platforms	Win ENT	Win STD	Win ESS	Mac ENT	Mac STD	Mac ESS	Linux ENT	Linux STD	Linux ESS
<b>PostgreSQL Objects Management</b>									
Support PostgreSQL Servers 7.3 or above	+	+	+	+	+	+	+	+	+
Create /drop databases and schemas	+	+	+	+	+	+	+	+	+
Support all PostgreSQL objects: tables, views and functions	+	+	+	+	+	+	+	+	+
Support all subobjects: fields, indexes, foreign keys, uniques, checks, rules and triggers	+	+	+	+	+	+	+	+	+
Managing tablespace	+	+	+	+	+	+	+	+	+
Managing cast and languages	+	+	+	+	+	+	+	+	+
Managing aggregates, conversions, domains, trigger functions, operators and classes, sequences and types	+	+	+	+	+	+	+	+	+
Support of character set and unicode	+	+	+	+	+	+	+	+	+
<b>Connection Management and Navigation</b>									
Multiple local/remote PostgreSQL server connections	+	+	+	+	+	+	+	+	+
Connection to PostgreSQL Server through SSH	+	+	+	+	+	+	+	+	+
Connection to PostgreSQL Server through HTTP	+	+	+	+	+	+	+	+	+
Support SSL secure connection for PostgreSQL 8.4 above	+	+	+	+	+	+	+	+	+
Keep connection alive	+	+	+	+	+	+	+	+	+
Export/Import connections settings	+	+	+	+	+	+	+	+	+
Customize selected databases to work with	+	+	+	+	+	+	+	+	+
<b>Data Viewer and Editor</b>									
UUID/GUID Generator	+	+	+	+	+	+	+	+	+
Grid view	+	+	+	+	+	+	+	+	+

Form view	+	+	-	+	+	-	+	+	-
Text, Hex and BLOB viewer/editor	+	+	+	+	+	+	+	+	+
Copying and pasting selected records	+	+	+	+	+	+	+	+	+
Customize display formats	+	+	+	+	+	+	+	+	+
Incremental search, sorting and formatting table grid	+	+	+	+	+	+	+	+	+
Support of raw mode edit: apply PostgreSQL built-in function	+	+	+	+	+	+	+	+	+
Support of foreign key data selection: locate the available values from the reference table	+	+	-	+	+	-	+	+	-
Filtering Records	+	+	-	+	+	-	+	+	-
Ability to set number of records showing on each page	+	+	+	+	+	+	+	+	+
<b>SQL Processing</b>									
Visual query/view builder to facilitate the process of building complicated queries	+	+	-	+	+	-	+	+	-
Ability to create parameter queries	+	+	+	+	+	+	+	+	+
Code completion for SQL editors	+	+	-	+	+	-	+	+	-
SQL Code Beautifier/Minifier	+	+	+	+	+	+	+	+	+
Multiple SQL editors with syntax highlighted feature	+	+	+	+	+	+	+	+	+
Support code folding, highlighted matches words and brackets	+	+	+	+	+	+	+	+	+
Copy SQL with quote	+	+	+	+	+	+	+	+	+
Word Wrap	+	+	+	+	+	+	+	+	+
Zoom In/Out for SQL editors	+	+	+	+	+	+	+	+	+
Enhanced find and replace options in SQL editors	+	+	+	+	+	+	+	+	+
Ability to preview SQL before execution	+	+	+	+	+	+	+	+	+
Support to return multi-resultsets	+	+	+	-	-	-	+	+	+
SQL console	+	+	+	+	+	+	-	-	-
<b>Data Modeling Tools</b>									
Reverse engineering	+	-	-	+	-	-	+	-	-

Support of several diagram notations like Crow's Foot, IDEF1x and UML	+	-	-	+	-	-	+	-	-
Ability to create/edit/drop database objects, visually set relationships between them	+	-	-	+	-	-	+	-	-
Support of auto layout	+	-	-	+	-	-	+	-	-
Support numerous visual customization options like adding vertices, layers, images and notes/labels	+	-	-	+	-	-	+	-	-
Zoom In/Out for ER diagram	+	-	-	+	-	-	+	-	-
Comparing and synchronizing database/model	+	-	-	+	-	-	+	-	-
Generating complex SQL/DDDL	+	-	-	+	-	-	+	-	-
Multi-page print preview	+	-	-	-	-	-	+	-	-
Ability to print the database diagram to PDF file	+	-	-	+	-	-	+	-	-
Ability to save the database diagram as a graphic file, i.e. PNG, SVG	+	-	-	+	-	-	+	-	-
<b>Diversified Import and Export Competence</b>									
Importing data from ODBC: MSSQL, Oracle etc	+	+	-	+	+	-	-	-	-
Importing data from MS Excel	+	+	-	+	+	-	-	-	-
Importing data from MS Access	+	+	-	-	-	-	-	-	-
Importing data from plain text file formats : TXT, CSV, XML etc	+	+	+	+	+	+	+	+	+
Importing data from other file formats : DBF etc	+	+	-	+	+	-	+	+	-
Exporting data to MS Excel	+	+	-	+	+	-	+	+	-
Exporting data to MS Access	+	+	-	-	-	-	-	-	-
Exporting data to HTML	+	+	-	-	-	-	+	+	-
Exporting data to plain text file formats : TXT, CSV, XML etc	+	+	+	+	+	+	+	+	+
Exporting data to most popular formats: DBF etc	+	+	-	+	+	-	+	+	-
<b>Other Data Manipulation Tools</b>									
Data transfer	+	+	-	+	+	-	+	+	-

Transferring data to SQL script as INSERT statement	+	+	-	+	+	-	+	+	-
Transferring data to a SQL file with designated SQL format and encoding	+	+	-	+	+	-	+	+	-
Data synchronization	+	+	-	+	+	-	+	+	-
Structure synchronization	+	+	-	+	+	-	+	+	-
<b>Backup/Restore</b>									
Backing up/restoring database	+	+	-	+	+	-	+	+	-
Backing up database with chosen database objects	+	+	-	+	+	-	+	+	-
Restoring database with chosen database objects	+	+	-	-	-	-	+	+	-
Converting backup file into SQL script	+	+	-	+	+	-	+	+	-
Compressing/Decompressing Backup files	+	+	-	-	-	-	+	+	-
<b>Schedule for Batch Job</b>									
Ability to set schedule on backup, query execution, import/export, data transfer and data synchronization	+	+	-	+	+	-	+	+	-
Schedule report printing	+	-	-	-	-	-	-	-	-
Schedule to print report to files: PDF, Excel, HTML etc	+	-	-	-	-	-	-	-	-
Sending notification e-mail for schedule task	+	+	-	+	+	-	+	+	-
Ability to attach Export Wizard/Report result files in Batch Job notification email	+	+	-	+	+	-	+	+	-
Notification email Support SSL/TLS	+	+	-	-	-	-	+	+	-
<b>Server Security and Maintenance Services</b>									
Visual user management	+	+	+	+	+	+	+	+	+
Visual privilege manager	+	+	+	+	+	+	+	+	+
Ability to duplicate user, role, etc	+	+	+	+	+	+	+	+	+
Analyzing, vacuuming and reindexing tables	+	+	+	+	+	+	+	+	+
Server Monitor to view system variables and server status information	+	+	-	+	+	-	+	+	-

Report Management Tools									
Report Builder	+	-	-	-	-	-	-	-	-
Report Viewer	+	-	-	-	-	-	-	-	-
Other Useful Features									
ER Diagram view	+	+	-	+	+	-	+	+	-
Ability to customize connection colorings	+	+	+	+	+	+	+	+	+
Support of database wide search	+	+	-	+	+	-	+	+	-
Objects search filter	+	+	+	+	+	+	+	+	+
Backing up schema/table to SQL script	+	+	+	+	+	+	+	+	+
Executing SQL script	+	+	+	+	+	+	+	+	+
Duplicate/empty/truncate tables	+	+	+	+	+	+	+	+	+
Create table/view shortcut on desktop	+	+	+	+	+	+	-	-	-
Ability to create Favorites list	+	+	-	+	+	-	+	+	-
Ability to set auto-save for Query, Function, etc	+	+	+	+	+	+	+	+	+
Print database/table Structure	+	+	-	+	+	-	+	+	-
Support of docking/tab windows	+	+	+	+	+	+	-	-	-
Support of virtual grouping: provide logical organization over connections and objects	+	+	-	+	+	-	+	+	-
Ability to run profile from command line	+	+	-	+	+	-	+	+	-
Log files: keep track on the actions have been performed in Navicat	+	+	+	+	+	+	+	+	+
Ability to copy Connection Information into Clipboard	+	+	+	+	+	+	+	+	+

## Navicat for SQLite

**Note:** ESS stands for Navicat Essentials.

Platforms	Win ENT	Win STD	Win ESS	Mac ENT	Mac STD	Mac ESS	Linux ENT	Linux STD	Linux ESS
<b>SQLite Objects Management</b>									
Support SQLite 2 and 3	+	+	+	+	+	+	+	+	+
Attach/detach databases	+	+	+	+	+	+	+	+	+
Support all SQLite objects: tables and views	+	+	+	+	+	+	+	+	+
Support all subobjects: triggers and indexes	+	+	+	+	+	+	+	+	+
Support to view master table	+	+	+	+	+	+	+	+	+
Support of character set and unicode	+	+	+	+	+	+	+	+	+
<b>Connection Management and Navigation</b>									
Multiple local/remote SQLite database connections	+	+	+	+	+	+	+	+	+
Support to connect a new or an existing database	+	+	+	+	+	+	+	+	+
Support encrypted SQLite file (wxSQLite3)	+	+	+	+	+	+	+	+	+
Connection to SQLite database through HTTP	+	+	+	+	+	+	+	+	+
Export/Import connections settings	+	+	+	+	+	+	+	+	+
<b>Data Viewer and Editor</b>									
Grid view	+	+	+	+	+	+	+	+	+
Form view	+	+	-	+	+	-	+	+	-
Text, Hex and BLOB viewer/editor	+	+	+	+	+	+	+	+	+
Copying and pasting selected records	+	+	+	+	+	+	+	+	+
Customize display formats	+	+	+	+	+	+	+	+	+
Incremental search, sorting and formatting table grid	+	+	+	+	+	+	+	+	+
Support of raw mode edit: apply SQLite built-in function	+	+	+	+	+	+	+	+	+
Ability to show/hide ROWID	+	+	+	+	+	+	+	+	+



Support of foreign key data selection: locate the available values from the reference table	+	+	-	+	+	-	+	+	-
Filtering Records	+	+	-	+	+	-	+	+	-
Ability to set number of records showing on each page	+	+	+	+	+	+	+	+	+
<b>SQL Processing</b>									
Visual query/view builder to facilitate the process of building complicated queries	+	+	-	+	+	-	+	+	-
Ability to create parameter queries	+	+	+	+	+	+	+	+	+
Code completion for SQL editors	+	+	-	+	+	-	+	+	-
SQL Code Beautifier/Minifier	+	+	+	+	+	+	+	+	+
Multiple SQL editors with syntax highlighted feature	+	+	+	+	+	+	+	+	+
Support code folding, highlighted matches words and brackets	+	+	+	+	+	+	+	+	+
Copy SQL with quote	+	+	+	+	+	+	+	+	+
Word Wrap	+	+	+	+	+	+	+	+	+
Zoom In/Out for SQL editors	+	+	+	+	+	+	+	+	+
Enhanced find and replace options in SQL editors	+	+	+	+	+	+	+	+	+
Ability to preview SQL before execution	+	+	+	+	+	+	+	+	+
Support to return multi-resultsets	+	+	+	-	-	-	+	+	+
SQL Console	+	+	-	+	+	-	-	-	-
<b>Data Modeling Tools</b>									
Reverse engineering	+	-	-	+	-	-	+	-	-
Support of several diagram notations like Crow's Foot, IDEF1x and UML	+	-	-	+	-	-	+	-	-
Ability to create/edit/drop database objects, visually set relationships between them	+	-	-	+	-	-	+	-	-
Support of auto layout	+	-	-	+	-	-	+	-	-
Support numerous visual customization options like adding vertices, layers, images and notes/labels	+	-	-	+	-	-	+	-	-

Zoom In/Out for ER diagram	+	-	-	+	-	-	+	-	-
Comparing and synchronizing database/model	+	-	-	+	-	-	+	-	-
Generating complex SQL/DDDL	+	-	-	+	-	-	+	-	-
Multi-page print preview	+	-	-	-	-	-	+	-	-
Ability to print the database diagram to PDF file	+	-	-	+	-	-	+	-	-
Ability to save the database diagram as a graphic file, i.e. PNG, SVG	+	-	-	+	-	-	+	-	-
<b>Diversified Import and Export Competence</b>									
Importing data from ODBC: MSSQL, Oracle etc	+	+	-	+	-	-	-	-	-
Importing data from MS Excel	+	+	-	+	+	-	-	-	-
Importing data from MS Access	+	+	-	-	-	-	-	-	-
Importing data from plain text file formats : TXT, CSV, XML etc	+	+	+	+	+	+	+	+	+
Importing data from other file formats : DBF etc	+	+	-	+	+	-	+	+	-
Exporting data to MS Excel	+	+	-	+	+	-	+	+	-
Exporting data to MS Access	+	+	-	-	-	-	-	-	-
Exporting data to HTML	+	+	-	-	-	-	+	+	-
Exporting data to plain text file formats : TXT, CSV, XML etc	+	+	+	+	+	+	+	+	+
Exporting data to most popular formats: DBF etc	+	+	-	+	+	-	+	+	-
<b>Other Data Manipulation Tools</b>									
Data transfer	+	+	-	+	+	-	+	+	-
Transferring data to SQL script as INSERT statement	+	+	-	+	+	-	+	+	-
Transferring data to a SQL file with designated SQL format and encoding	+	+	-	+	+	-	+	+	-
Data synchronization	+	+	-	+	+	-	+	+	-
<b>Backup/Restore</b>									
Backing up/restoring database	+	+	-	+	+	-	+	+	-
Backing up database with chosen database objects	+	+	-	+	+	-	+	+	-

Restoring database with chosen database objects	+	+	-	-	-	-	+	+	-
Converting backup file into SQL script	+	+	-	+	+	-	+	+	-
Compressing/Decompressing Backup files	+	+	-	-	-	-	+	+	-
<b>Schedule for Batch Job</b>									
Ability to set schedule on backup, query execution, import/export, data transfer and data synchronization	+	+	-	+	+	-	+	+	-
Schedule report printing	+	-	-	-	-	-	-	-	-
Schedule to print report to files: PDF, Excel, HTML etc	+	-	-	-	-	-	-	-	-
Sending notification e-mail for schedule task	+	+	-	+	+	-	+	+	-
Ability to attach Export Wizard/Report result files in Batch Job notification email	+	+	-	+	+	-	+	+	-
Notification email Support SSL/TLS	+	+	-	-	-	-	+	+	-
<b>Server Security and Maintenance Services</b>									
Analyzing, re-indexing tables	+	+	+	+	+	+	+	+	+
<b>Report Management Tools</b>									
Report Builder	+	-	-	-	-	-	-	-	-
Report Viewer	+	-	-	-	-	-	-	-	-
<b>Other Useful Features</b>									
ER Diagram view	+	+	-	+	+	-	+	+	-
Ability to customize connection colorings	+	+	+	+	+	+	+	+	+
Support of database wide search	+	+	-	+	+	-	+	+	-
Objects search filter	+	+	+	+	+	+	+	+	+
Backing up database/table to SQL script	+	+	+	+	+	+	+	+	+
Executing SQL script	+	+	+	+	+	+	+	+	+
Duplicate/empty/truncate tables	+	+	+	+	+	+	+	+	+
Create table/view shortcut on desktop	+	+	+	+	+	+	-	-	-
Ability to create Favorites list	+	+	-	+	+	-	+	+	-
Ability to set auto-save for Query, etc	+	+	+	+	+	+	+	+	+
Print database/table Structure	+	+	-	+	+	-	+	+	-

Support of docking/tab windows	+	+	+	+	+	+	-	-	-
Support of virtual grouping: provide logical organization over connections and objects	+	+	-	+	+	-	+	+	-
Ability to run profile from command line	+	+	-	+	+	-	+	+	-
Log files: keep track on the actions have been performed in Navicat	+	+	+	+	+	+	+	+	+
Ability to copy Connection Information into Clipboard	+	+	+	+	+	+	+	+	+

## Navicat for SQL Server

**Note:** ESS stands for Navicat Essentials.

Platforms	Win ENT	Win STD	Win ESS	Mac ENT	Mac STD	Mac ESS
<b>SQL Server Objects Management</b>						
Support SQL Server 2000 or above	+	+	+	+	+	+
Support SQL Azure	+	+	+	+	+	+
Create /drop databases and schemas	+	+	+	+	+	+
Support all SQL Server objects: tables, views and functions	+	+	+	+	+	+
Support all subobjects: fields, indexes, foreign keys, uniques, checks and triggers	+	+	+	+	+	+
Managing linked servers, server triggers	+	+	+	+	+	+
Managing assemblies, partition functions, partition schemes and database triggers	+	+	+	+	+	+
Managing indexes, synonyms and triggers	+	+	+	+	+	+
Support of character set and unicode	+	+	+	+	+	+
<b>Connection Management and Navigation</b>						
Multiple local/remote SQL Server server connections	+	+	+	+	+	+
Connection to SQL Server through SSH	+	+	+	+	+	+
Keep connection alive	+	+	+	+	+	+
Export/Import connections settings	+	+	+	+	+	+
<b>Data Viewer and Editor</b>						
UUID/GUID Generator	+	+	+	+	+	+
Grid view	+	+	+	+	+	+
Form view	+	+	-	+	+	-
Text, Hex and BLOB viewer/editor	+	+	+	+	+	+
Copying and pasting selected records	+	+	+	+	+	+
Customize display formats	+	+	+	+	+	+
Incremental search, sorting and formatting table grid	+	+	+	+	+	+
Support of raw mode edit: apply SQL Server built-in function	+	+	+	+	+	+
Support of foreign key data selection: locate the available values from the reference table	+	+	-	+	+	-
Filtering Records	+	+	-	+	+	-

Ability to set number of records showing on each page	+	+	+	+	+	+
<b>SQL Processing</b>						
Visual query/view builder to facilitate the process of building complicated queries	+	+	-	+	+	-
Ability to create parameter queries	+	+	+	+	+	+
Code completion for SQL editors	+	+	-	+	+	-
Code outline	+	+	-	+	+	-
SQL Code Beautifier/Minifier	+	+	+	+	+	+
Multiple SQL editors with syntax highlighted feature	+	+	+	+	+	+
Support code folding, highlighted matches words and brackets	+	+	+	+	+	+
Copy SQL with quote	+	+	+	+	+	+
Word Wrap	+	+	+	+	+	+
Zoom In/Out for SQL editors	+	+	+	+	+	+
Enhanced find and replace options in SQL editors	+	+	+	+	+	+
Ability to preview SQL before execution	+	+	+	+	+	+
Support to return multi-resultsets	+	+	+	-	-	-
SQL console	+	+	+	+	+	+
<b>Data Modeling Tools</b>						
Reverse engineering	+	-	-	+	-	-
Support of several diagram notations like Crow's Foot, IDEF1x and UML	+	-	-	+	-	-
Ability to create/edit/drop database objects, visually set relationships between them	+	-	-	+	-	-
Support of auto layout	+	-	-	+	-	-
Support numerous visual customization options like adding vertices, layers, images and notes/labels	+	-	-	+	-	-
Zoom In/Out for ER diagram	+	-	-	+	-	-
Comparing and synchronizing database/model	+	-	-	+	-	-
Generating complex SQL/DDDL	+	-	-	+	-	-
Multi-page print preview	+	-	-	-	-	-
Ability to print the database diagram to PDF file	+	-	-	+	-	-
Ability to save the database diagram as a graphic file, i.e. PNG, SVG	+	-	-	+	-	-
<b>Diversified Import and Export Competence</b>						
Importing data from ODBC: MSSQL, Oracle etc	+	+	-	+	+	-

Importing data from MS Excel	+	+	-	+	+	-
Importing data from MS Access	+	+	-	-	-	-
Importing data from plain text file formats : TXT, CSV, XML etc	+	+	+	+	+	+
Importing data from other file formats : DBF etc	+	+	-	+	+	-
Exporting data to MS Excel	+	+	-	+	+	-
Exporting data to MS Access	+	+	-	-	-	-
Exporting data to HTML	+	+	-	-	-	-
Exporting data to plain text file formats : TXT, CSV, XML etc	+	+	+	+	+	+
Exporting data to most popular formats: DBF etc	+	+	-	+	+	-
<b>Other Data Manipulation Tools</b>						
Data transfer	+	+	-	+	+	-
Transferring data to SQL script as INSERT statement	+	+	-	+	+	-
Transferring data to a SQL file with designated SQL format and encoding	+	+	-	+	+	-
Data synchronization	+	+	-	+	+	-
Structure synchronization	+	+	-	+	+	-
<b>Schedule for Batch Job</b>						
Ability to set schedule on query execution, import/export, data transfer and data synchronization	+	+	-	+	+	-
Schedule report printing	+	-	-	-	-	-
Schedule to print report to files: PDF, Excel, HTML etc	+	-	-	-	-	-
Sending notification e-mail for schedule task	+	+	-	+	+	-
Ability to attach Export Wizard/Report result files in Batch Job notification email	+	+	-	+	+	-
Notification email Support SSL/TLS	+	+	-	-	-	-
<b>Server Security and Maintenance Services</b>						
Visual user management	+	+	+	+	+	+
Visual privilege manager	+	+	+	+	+	+
Ability to duplicate user, role, etc	+	+	+	+	+	+
Maintenance database/schema objects	+	+	+	+	+	+
Server Monitor to view system variables and server status information	+	+	-	+	+	-
<b>Report Management Tools</b>						
Report Builder	+	-	-	-	-	-
Report Viewer	+	-	-	-	-	-

Other Useful Features						
ER Diagram view	+	+	-	+	+	-
Ability to customize connection colorings	+	+	+	+	+	+
Support of database wide search	+	+	-	+	+	-
Objects search filter	+	+	+	+	+	+
Backing up schema/table to SQL script	+	+	+	+	+	+
Executing SQL script	+	+	+	+	+	+
Duplicate/empty tables	+	+	+	+	+	+
Create table/view shortcut on desktop	+	+	+	+	+	+
Ability to create Favorites list	+	+	-	+	+	-
Ability to set auto-save for Query, Function, etc	+	+	+	+	+	+
Print database/table Structure	+	+	-	+	+	-
Support of docking/tab windows	+	+	+	+	+	+
Support of virtual grouping: provide logical organization over connections and objects	+	+	-	+	+	-
Ability to run profile from command line	+	+	-	+	+	-
Log files: keep track on the actions have been performed in Navicat	+	+	+	+	+	+
Ability to copy Connection Information into Clipboard	+	+	+	+	+	+



## Registration

To make it economic and efficient for you to purchase our products, over 95% of customers order Navicat via our [Online Shop](#) using major Credit Cards - MasterCard, Visa, Euro card, JCB and American Express. All Online orders are processed by **Share-it!** and **worldpay**. The VeriSign Certificate for SSL transactions provided will ensure you a secured Online transactions.

If you have ordered Navicat software and would like to review your order information, or if you have questions about ordering, payments, or shipping procedures, please contact our [Navicat Sales Department](#).

After purchase you will obtain a **Registration Key** to activate your licensed Navicat by e-mail within 24 hours after we received your order. Please make sure to enter a valid e-mail address in your order. If you have not received the keys within 24 hours, it is probably that the e-mail we sent was blocked by your email spam filter. To resend your download information and keys, please submit your registered email address to our [Customer Center](#). If you get no reply from the resend form, please contact our [Navicat Sales Department](#).

Besides, if you feel uncomfortable with providing your personal information over the Internet, we accept Purchase Order and Bank/Wire Transfer. Please visit our [Offline Order](#).

## Installation

We strongly suggest that you shut down any opened applications. This will help ensure a smooth installation.

**Note:** Installing Navicat does not include the server installation. You should download and install the server manually.

**Note:** For user who has been trying our unregistered version, just simply key in the **Registration Key** (16 digit) on the pop up Registration screen.

### Installation for Online Version

1. Open or Save the **.exe** file.
2. Click **Next** at the Welcome Screen.
3. Read the License Agreement. Accept it and click **Next**.
4. Accept the location of the program icons by clicking Next. If you wish to change the destination of the folder for Navicat program click Browse.
5. Follow the remaining steps.
6. After installed, key in the **Registration Key** (16 digit) on the pop up Registration screen.

### Installation for CD Version

1. Load the Navicat CD Installation disk into the CD-ROM drive.
2. Open the **.exe** file.
3. Click **Next** at the Welcome Screen.
4. Read the License Agreement. Accept it and click **Next**.
5. Accept the location of the program icons by clicking Next. If you wish to change the destination of the folder for Navicat program click Browse.
6. Follow the remaining steps.
7. After installed, key in the **Registration Key** (16 digit) on the pop up Registration screen.

## Migrate Navicat to new computer

All your connection settings are stored in **registry**. To view the registry record, in Windows, Choose Start -> Run, then type "regedit".

Navicat version	Path
Navicat Premium	HKEY_CURRENT_USER/Software/PremiumSoft/NavicatPremium
	HKEY_CURRENT_USER/Software/PremiumSoft/Navicat
	HKEY_CURRENT_USER/Software/PremiumSoft/NavicatOra
	HKEY_CURRENT_USER/Software/PremiumSoft/NavicatPG
	HKEY_CURRENT_USER/Software/PremiumSoft/NavicatSQLite
	HKEY_CURRENT_USER/Software/PremiumSoft/NavicatMSSQL
Navicat for MySQL	HKEY_CURRENT_USER/Software/PremiumSoft/Navicat
Navicat for Oracle	HKEY_CURRENT_USER/Software/PremiumSoft/NavicatOra
Navicat for PostgreSQL	HKEY_CURRENT_USER/Software/PremiumSoft/NavicatPG
Navicat for SQLite	HKEY_CURRENT_USER/Software/PremiumSoft/NavicatSQLite
Navicat for SQL Server	HKEY_CURRENT_USER/Software/PremiumSoft/NavicatMSSQL

1. In Navicat, choose File -> Export Connections. The exported file (.ncx) contains all your connection settings.
2. Backup the exported file (.ncx).
3. Uninstall Navicat from the existing computer.
4. Re-install Navicat in the new computer.
5. Open Navicat and choose File -> Import Connections in the new computer.

When a new connection being established, Navicat will create a subfolder under the **Settings Save Path**. Most files are stored within this subfolder. To look for the path, right click the connection and choose Connection Properties -> Advanced -> Settings Save Path.

Moreover, all your saved profiles are stored under **profiles**. To look for the path, choose Tools -> Options -> Miscellaneous -> Profiles Save Path.

## **Maintenance/Upgrade**

### **How to purchase the maintenance plan?**

Navicat Software Maintenance Plan allows Navicat users to receive priority email support, receiving software upgrades and receiving bug fix releases at no additional cost during the protected period.

Subscription to the Maintenance Plan is done at the time of your software license purchase or within 90 days as of your purchase date - it cannot be added to a previously purchased product at a later date.

For details, please [click here](#).

### **How to upgrade your Navicat?**

If you want to upgrade installed copy of Navicat to the latest release, please submit your registered email address on the [Customer Center](#).

Please install the latest version into current Navicat Installation folder, it will replace your previous Navicat. But your current settings will remain unchanged.

## End-User License Agreement

IMPORTANT: THIS SOFTWARE END USER LICENSE AGREEMENT ("EULA") IS A LEGAL AGREEMENT BETWEEN YOU (EITHER AN INDIVIDUAL OR, IF PURCHASED OR OTHERWISE ACQUIRED BY OR FOR AN ENTITY, AN ENTITY) AND PREMIUMSOFT CYBERTECH LTD..READ IT CAREFULLY BEFORE COMPLETING THE INSTALLATION PROCESS AND USING THE SOFTWARE. IT PROVIDES A LICENSE TO USE THE SOFTWARE AND CONTAINS WARRANTY INFORMATION AND LIABILITY DISCLAIMERS. BY INSTALLING AND USING THE SOFTWARE, YOU ARE CONFIRMING YOUR ACCEPTANCE OF THE SOFTWARE AND AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO BE BOUND BY THESE TERMS, THEN DO NOT INSTALL THE SOFTWARE AND RETURN THE SOFTWARE TO YOUR PLACE OF PURCHASE. THIS EULA SHALL APPLY ONLY TO THE SOFTWARE SUPPLIED BY PREMIUMSOFT CYBERTECH LTD. HEREWITH REGARDLESS OF WHETHER OTHER SOFTWARE IS REFERRED TO OR DESCRIBED HEREIN.

### 1. Definitions

- a. "Non-commercial Version" means a version of the Software, so identified, for use by i) the individual who is a natural person and not a corporation, company, partnership or association or other entity or organization (ii) the individual who is a student, faculty or staff member at an educational institution, and (iii) staff of a non-profit organization or charity organization only. For purposes of this definition, "educational institution" means a public or private school, college, university and other post secondary educational establishment. A non-profit organization is an organization whose primary objective is to support an issue or matter of private interest or public concern for non-commercial purposes.
- b. "Not For Resale (NFR) Version" means a version, so identified, of the Software to be used to review and evaluate the Software, only.
- c. "PremiumSoft" means PREMIUMSOFT CYBERTECH LTD. and its licensors, if any.
- d. "Software" means only the PremiumSoft software program(s) and third party software programs, in each case, supplied by PremiumSoft herewith, and corresponding documentation, associated media, printed materials, and online or electronic documentation.

- e. "Unregistered version", "Trial version" or "Demo version" means an unregistered copy of the SOFTWARE ("UNREGISTERED SOFTWARE") which may be used by the USER for evaluation purposes for a period of thirty (30) days following the initial installation of the UNREGISTERED SOFTWARE. At the end of the trial period ("TRIAL PERIOD"), the USER must either register the SOFTWARE or remove it from his system. The UNREGISTERED SOFTWARE may be freely copied and distributed to other users for their evaluation.
- f. "Navicat Essentials" means a version of the Software, so identified, to be used for commercial purpose.

## 2. License Grants

The licenses granted in this Section 2 are subject to the terms and conditions set forth in this EULA:

- a. Subject to Section 2(b), you may install and use the Software on a single computer; OR install and store the Software on a storage device, such as a network server, used only to install the Software on your other computers over an internal network, provided you have a license for each separate computer on which the Software is installed and run. Except as otherwise provided in Section 2(b), a license for the Software may not be shared, installed or used concurrently on different computers.
- b. In addition to the single copy of the Software permitted in Section 2(a), the primary user of the computer on which the Software is installed may make a second copy of the Software and install it on either a portable computer or a computer located at his or her home for his or her exclusive use, provided that:
  - A. the second copy of the Software on the portable or home computer (i) is not used at the same time as the copy of the Software on the primary computer and (ii) is used by the primary user solely as allowed for such version or edition (such as for educational use only),
  - B. the second copy of the Software is not installed or used after the time such user is no longer the primary user of the primary computer on which the Software is installed.

- c. In the event the Software is distributed along with other PremiumSoft software products as part of a suite of products (collectively, the "Studio"), the license of the Studio is licensed as a single product and none of the products in the Studio, including the Software, may be separated for installation or use on more than one computer.
- d. You may make one copy of the Software in machine-readable form solely for backup purposes. You must reproduce on any such copy all copyright notices and any other proprietary legends on the original copy of the Software. You may not sell or transfer any copy of the Software made for backup purposes.
- e. You agree that PremiumSoft may audit your use of the Software for compliance with these terms at any time, upon reasonable notice. In the event that such audit reveals any use of the Software by you other than in full compliance with the terms of this Agreement, you shall reimburse PremiumSoft for all reasonable expenses related to such audit in addition to any other liabilities you may incur as a result of such non-compliance.
- f. Your license rights under this EULA are non-exclusive.

### **3. License Restrictions**

- a. Other than as set forth in Section 2, you may not make or distribute copies of the Software, or electronically transfer the Software from one computer to another or over a network.
- b. You may not alter, merge, modify, adapt or translate the Software, or decompile, reverse engineer, disassemble, or otherwise reduce the Software to a human-perceivable form.
- c. Unless otherwise provided herein, you may not rent, lease, or sublicense the Software.
- d. Other than with respect to a Trial / Demo Version, Non-commercial Lite Version or a Not For Resale Version of the Software, you may permanently transfer all of your rights under this EULA only as part of a sale or transfer, provided you retain no copies, you transfer all of the Software (including all component parts, the media and printed materials, any upgrades, this EULA, the serial numbers, and, if applicable, all other software products provided together with the Software), and the recipient agrees to the terms of this EULA. If the Software is an upgrade, any transfer must include all prior versions of the Software from which you are upgrading. If the copy of the Software is licensed as part of the whole Studio (as defined above), the Software shall be transferred only with and as part of the sale or transfer of the whole Studio, and not separately. You may retain no copies of the Software. You may not sell or transfer any Trial / Demo Version, Non-commercial Lite Version or Not For Resale Version of the Software.



- e. Unless otherwise provided herein, you may not modify the Software or create derivative works based upon the Software.
- f. Non-commercial Versions of the Software may not be used for, or distributed to any party for, any commercial purpose.
- g. Unless otherwise provided herein, you shall not
  - A. in the aggregate, install or use more than one copy of the Trial / Demo Version and Non-commercial Lite Version of the Software,
  - B. download the Trial / Demo Version and Non-commercial Lite Version of the Software under more than one username,
  - C. alter the contents of a hard drive or computer system to enable the use of the Trial / Demo Version of the Software for an aggregate period in excess of the trial period for one license to such Trial / Demo Version,
  - D. disclose the results of software performance benchmarks obtained using the Trial / Demo Version or Non-commercial Lite Version to any third party without PremiumSoft prior written consent, or
  - E. use the Trial / Demo Version of the Software for a purpose other than the sole purpose of determining whether to purchase a license to a commercial or education version of the software; provided, however, notwithstanding the foregoing, you are strictly prohibited from installing or using the Trial / Demo Version or Non-commercial Lite Version of the Software for any commercial training purpose.
- h. You may only use the Not for Resale Version of the Software to review and evaluate the Software.
- i. You may receive the Software in more than one medium but you shall only install or use one medium. Regardless of the number of media you receive, you may use only the medium that is appropriate for the server or computer on which the Software is to be installed.
- j. You may receive the Software in more than one platform but you shall only install or use one platform.
- k. You shall not use the Software to develop any application having the same primary function as the Software.
- l. In the event that you fail to comply with this EULA, PremiumSoft may terminate the license and you must destroy all copies of the Software (with all other rights of both parties and all other provisions of this EULA surviving any such termination).
- m. This program may include Oracle Instant Client (OCI). You agree that you shall
  - 1. not use of the Oracle Instant Client to the business operations;
  - 2. not assign, give, or transfer the Oracle Instant Client or an interest in them to another individual or entity;



- a. make the Programs available in any manner to any third party for use in the third party's business operations; and
  - b. title to the Programs from passing to the end user or any other party;
3. not reverse engineer, disassemble or decompilation the Oracle Instant Client and duplicate the Programs except for a sufficient number of copies of each Program for your licensed use and one copy of each Program media;
4. discontinue use and destroy or return to all copies of the Oracle Instant Client and documentation after termination of the Agreement;
5. not publish any results of benchmark tests run on the Programs;
6. comply fully with all relevant export laws and regulations of the United States and other applicable export and import laws to assure that neither the Oracle Instant Client, nor any direct product thereof, are exported, directly or indirectly, in violation of applicable laws;
7. allow PremiumSoft to audit your use of the Oracle Instant Client;

## **4. Upgrades**

If this copy of the Software is an upgrade from an earlier version of the Software, it is provided to you on a license exchange basis. You agree by your installation and use of such copy of the Software to voluntarily terminate your earlier EULA and that you will not continue to use the earlier version of the Software or transfer it to another person or entity unless such transfer is pursuant to Section 3.

## **5. Ownership**

The foregoing license gives you limited license to use the Software. PremiumSoft and its suppliers retain all rights, title and interest, including all copyright and intellectual property rights, in and to, the Software (as an independent work and as an underlying work serving as a basis for any application you may develop), and all copies thereof. All rights not specifically granted in this EULA, including Federal and International Copyrights, are reserved by PremiumSoft and its suppliers.

## 6. LIMITED WARRANTY AND DISCLAIMER

- a. Except with respect to Trial / Demo Version, Non-commercial Lite Version and Not For Resale Version of the Software, PremiumSoft warrants that, for a period of thirty (30) days from the date of delivery (as evidenced by a copy of your receipt): the physical media on which the Software is furnished will be free from defects in materials and workmanship under normal use. The Software is provided "as is". PremiumSoft makes no warranties, express or implied, arising from course of dealing or usage of trade, or statutory, as to any matter whatsoever.
- b. PremiumSoft provides no remedies or warranties, whether express or implied, for Trial / Demo version, Non-commercial Lite version and the Not for Resale version of the Software. Trial / Demo version, Non-commercial Lite version and the Not for Resale version of the Software are provided "as is".
- c. Except as set Forth in the foregoing limited warranty with respect to software other than Trial/ Demo version, Non-commercial Lite version and Not for Resale version, PremiumSoft and its suppliers disclaim all other warranties and representations, whether express, implied, or otherwise, including the warranties of merchantability or fitness for a particular purpose. Also, there is no warranty of non-infringement and title or quiet enjoyment. PremiumSoft does not warrant that the Software is error-free or will operate without interruption. The Software is not designed, intended or licensed for use in hazardous environments requiring fail-safe controls, including without limitation, the design, construction, maintenance or operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, and life support or weapons systems. PremiumSoft specifically disclaims any express or implied warranty of fitness for such purposes.
- d. If applicable law requires any warranties with respect to the Software, all such warranties are limited in duration to thirty (30) days from the date of delivery.
- e. No oral or written information or advice given by PremiumSoft, its dealers, distributors, agents or employees shall create a warranty or in any way increase the scope of ANY warranty PROVIDED HEREIN.

## **7. LIMITATION OF LIABILITY**

(a) Neither PremiumSoft nor its suppliers shall be liable to you or any third party for any indirect, special, incidental, punitive or consequential damages (including, but not limited to, damages for the inability to use equipment or access data, loss of business, loss of profits, business interruption or the like), arising out of the use of, or inability to use, the Software and based on any theory of liability including breach of contract, breach of warranty, tort (including negligence), product liability or otherwise, even if PremiumSoft or its representatives have been advised of the possibility of such damages.

## **8. Third Party Software**

The Software may contain third party software which requires notices and/or additional terms and conditions. By accepting this EULA, you are also accepting the additional terms and conditions of the third party software.

## **9. General**

No PremiumSoft dealer, agent or employee is authorized to make any amendment to this EULA.

This EULA contains the complete agreement between the parties with respect to the subject matter hereof, and supersedes all prior or contemporaneous agreements or understandings, whether oral or written. You agree that any varying or additional terms contained in any purchase order or other written notification or document issued by you in relation to the Software licensed hereunder shall be of no effect. The failure or delay of PremiumSoft to exercise any of its rights under this EULA or upon any breach of this EULA shall not be deemed a waiver of those rights or of the breach.

If any provision of this EULA shall be held by a court of competent jurisdiction to be contrary to law, that provision will be enforced to the maximum extent permissible, and the remaining provisions of this EULA will remain in full force and effect.

## **10. Basis of Bargain**

The Limited Warranty and Disclaimer and Limited Liability set forth above are fundamental elements of the basis of the agreement between PremiumSoft and you. PremiumSoft would not be able to provide the Software on an economic basis without such limitations. Such Limited Warranty and Disclaimer and Limited Liability inure to the benefit of PremiumSoft's licensors.

## 11. Term

By downloading and/or installing this SOFTWARE, the Licensor agrees to the terms of this EULA.

This license is effective until terminated. Licensor has the right to terminate your License immediately if you fail to comply with any term of this License.

"as is". Licensor makes no warranties, express or implied, arising from course of dealing or usage of trade, or statutory, as to any matter whatsoever. In particular, any and all warranties or merchantability, fitness for a particular purpose or non-infringement of third party rights are expressly excluded.

## 12. Governing Law

This License will be governed by the laws in force in Hong Kong. You hereby consent to the non-exclusive jurisdiction and venue sitting in Hong Kong to resolve any disputes arising under this EULA.

Should you have any questions concerning the validity of this License, please contact: [licensing@navicat.com](mailto:licensing@navicat.com). If you desire to contact the Licensor for any other reason, please contact [support@navicat.com](mailto:support@navicat.com).

PremiumSoft and other trademarks contained in the Software are trademarks or registered trademarks of PremiumSoft CyberTech Ltd. in the United States and/or other countries. Third party trademarks, trade names, product names and logos may be the trademarks or registered trademarks of their respective owners. You may not remove or alter any trademark, trade names, product names, logo, copyright or other proprietary notices, legends, symbols or labels in the Software. This EULA does not authorize you to use PremiumSoft or its licensors names or any of their respective trademarks.

## Getting Started

To start working with MySQL/Oracle/PostgreSQL/SQLite/SQL Server databases in Navicat, you should first establish a connection or several connections using the Connection Windows. If you are new to the server or 'Net in general' and are not quite sure how things work, you may want to look at:

- [MySQL User Manual](#)
- [Oracle Database Documentation](#)
- [PostgreSQL User Manual](#)
- [SQLite User Manual](#)
- [SQL Server MSDN Library](#)

Click  or choose File ->  **New Connection** to set up the Connection Properties.

- [Connection Settings](#)

After the connections being established, you can connect to database, manage its objects, table data, and so on. See the instructions below to learn how to perform these operations in the easiest way.

- [Working with databases/schemas](#)
- [Working with database/schema objects](#)

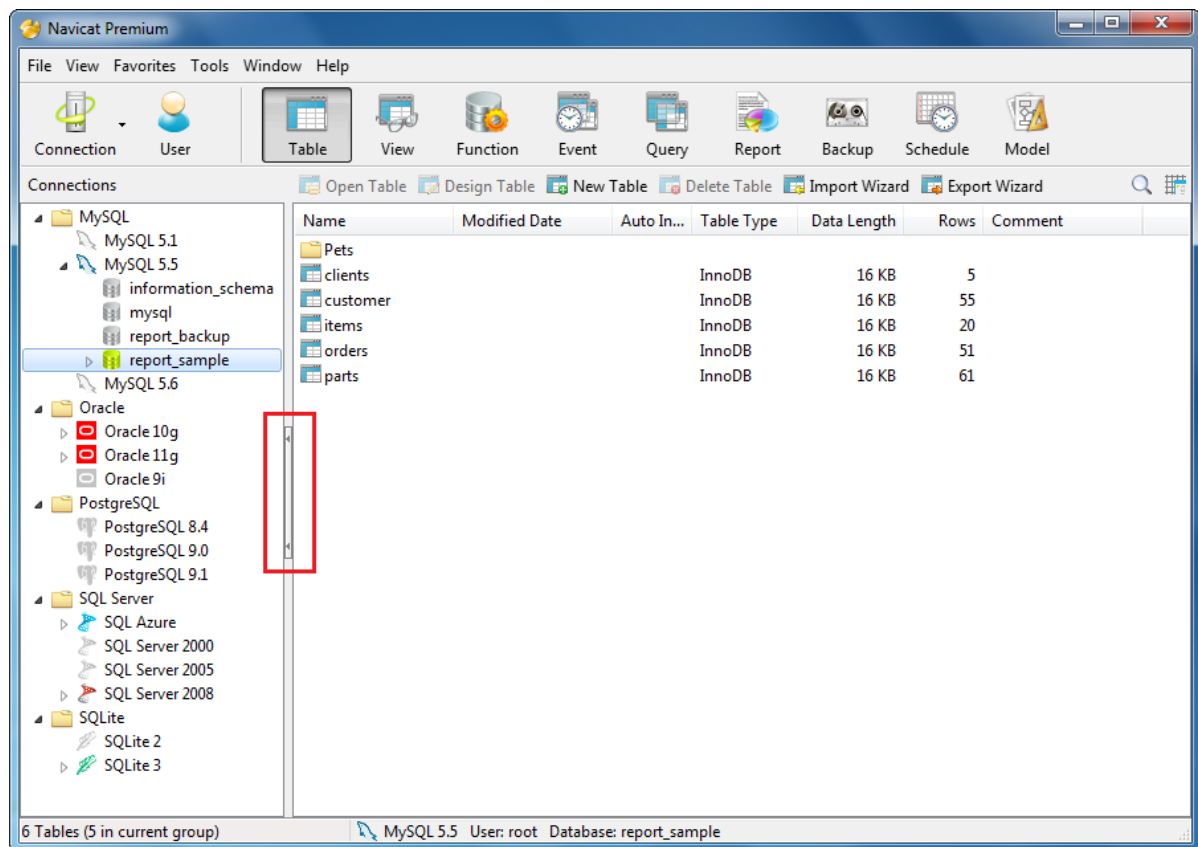
## Navicat Explorer!

The Navicat window includes a navigation pane (the left pane) and an object pane (the right pane).

Navigation Pane is the basic way to navigate with connections, databases and database objects. It employs tree structure which allows you to take action upon the database and their objects through their popup menus quickly and easily.

Object pane displays the opened tables, queries and so on. Toolbars at the top of window provide other controls that you can use to manipulate your data.



To view or hide Navigation Pane, click the **Red Indicator** below or choose View -> **Connection Tree** from main menu.



## Connection Settings

Navicat assembles utilitarian tools to manage your databases. To start managing your databases in Navicat, the first thing you require to do is to establish your server connection.

### Create Connection

Navicat provides three typical approaches to establish your connection, click  or choose File ->  **New Connection** to start the setup.

- [General Settings for MySQL](#)
- [General Settings for Oracle](#)
- [General Settings for PostgreSQL](#)
- [General Settings for SQLite](#)
- [General Settings for SQL Server](#)
- [SSH Settings](#) (Available only for MySQL, Oracle, PostgreSQL and SQL Server)
- [HTTP Settings](#) (Available only for MySQL, PostgreSQL and SQLite)

**Note:** For MySQL or PostgreSQL server, a commonly-used protocol - **Secure Sockets Layer (SSL)** is employed for managing the security of a message transmission on the Internet (see [SSL Settings](#) for details).

Navicat provides evaluated accounts for testing purpose.

The remote MySQL server connection settings are:


- Host Name/IP Address: server1.navicat.com
- Port: 4406
- User Name: navicat
- Password: testnavicat

The remote PostgreSQL server connection settings are:

- Host Name/IP Address: server1.navicat.com
- Port: 5432
- Initial Database: HR
- User Name: navicat
- Password: testnavicat


**Note:** Navicat authorizes you to make connection to remote server running on different platform, i.e. Windows, Mac, Linux and UNIX.

To create a new connection with the same properties as one of the existing connection has

- Right-click the connection in the navigation pane and choose  **Duplicate Connection....**
- The newly created connection will be named as "connectionname\_copy".

## Delete Connection

To delete a connection

- Right-click the connection in the navigation pane and choose  **Delete Connection.**
- Confirm deleting in the dialog window.


## Open Connection

To open a connection

- Double-click the connection to open in the navigation pane.


## Close Connection

To close a connection

- Right-click the connection in the navigation pane and choose  **Close Connection.**

## Edit Connection

To edit a connection information

- Close the connection if it is being opened.
- Right-click the connection and choose  **Connection Properties....**

## Open Connection Settings Save Path

To open a connection settings save path

- Select the connection in the navigation pane.
- Right-click the connection and choose **Go to settings save path** or press **Ctrl+G** to open the settings save path folder.



## Export Connection Settings

To export connection settings

- Choose File -> **Export Connections....**
- Select the connections and the export file path.

## Import Connection Settings

To import connection settings

- Choose File -> **Import Connections....**
- Specify the connection settings file path.
- Confirm replacing or skipping in the dialog window if the connection already exists.

## Achieve Connection Information

To achieve a connection information

- Open the connection in the navigation pane.
- Right-click the opened connection and choose **Connection Information....**

## General Settings for MySQL

The following instruction guides you through the process of creating a new connection. To successfully establish a new connection to local/remote MySQL - no matter via SSL, SSH or HTTP, set the connection properties in the corresponding boxes: Connection name, Host name, Port number, User name, and Password.

By default, MySQL gives "root" as username and leave the password field blank.

### Connection Name

A friendly name to best describe your connection.

### Host Name/IP Address

A host name where the database is situated or the IP address of the server.

### Port

A TCP/IP port for connecting to the database server.

### User Name

User name for connecting to the database server.

### Password

Password for connecting to the server.

You can connect to your MySQL Server remotely however for security reasons native remote direct connections to the MySQL server are disabled. Therefore, you cannot use Navicat Premium or other similar MySQL admin applications running on your computer to connect to the remote server directly unless the User Privileges has been configured.

If your Internet Service Provider (ISP) does not provide direct access to its server, Secure Tunneling Protocol (SSH) / HTTP is another solution.

## General Settings for Oracle

The following instruction guides you through the process of creating a new connection for server. To successfully establish a new connection to local/remote Oracle - no matter via SSH, set the connection properties in the corresponding boxes: Connection name, Host name, Port number, User name, and Password.

By default, Oracle created a number of user accounts upon installation. Administrative accounts: SYS, SYSTEM, SYSMAN, and DBSNMP. Sample schema accounts: SCOTT, HR, OE, OC, PM, IX and SH.

Navicat supports 2 types of Oracle Server connection:

- [Basic Connection](#)
- [TNS Connection](#)

If your Internet Service Provider (ISP) does not provide direct access to its server, Secure Tunneling Protocol (SSH) is another solution.

## Oracle Basic Connection General Settings

### Connection Name

A friendly name to best describe your connection.

### Connection Type

Connection type for connecting to the server: **Basic** or TNS.

#### Basic

In Basic mode, Navicat connects to Oracle through the Oracle Call Interface (OCI). OCI is an application programming interface that allows an application developer to use a third-generation language's native procedure or function calls to access the Oracle database server and control all phases of SQL statement execution. OCI is a library of standard database access and retrieval functions in the form of a dynamic-link library.

### Host Name/IP Address

A host name where the database is situated or the IP address of the server.

### Port

A TCP/IP port for connecting to the database server.

### Service Name/SID

Set the Service Name/SID which the user connects when making connection. Select the corresponding radio button.

### User Name

User name for connecting to the database server.

### Password

Password for connecting to the server.

## Oracle TNS Connection General Settings

### Connection Name

A friendly name to best describe your connection.

### Connection Type

Connection type for connecting to the server: Basic or **TNS**.

#### **TNS**

In TNS mode, Navicat connects to Oracle server using an alias entry from a tnsnames.ora file through the Oracle Call Interface (OCI). OCI is an application programming interface that allows an application developer to use a third-generation language's native procedure or function calls to access the Oracle database server and control all phases of SQL statement execution. OCI is a library of standard database access and retrieval functions in the form of a dynamic-link library.

### Net Service Name

The net service name.

### User Name

User name for connecting to the database server.

### Password

Password for connecting to the server.

## General Settings for PostgreSQL

The following instruction guides you through the process of creating a new connection. To successfully establish a new connection to local/remote PostgreSQL - no matter via SSH, HTTP or SSL, set the connection properties in the corresponding boxes: Connection name, Host name, Port number, Initial Database, User name, and Password.

By default, PostgreSQL gives "postgres" as username and leave the password field blank.

### Connection Name

A friendly name to best describe your connection.

### Host Name/IP Address

A host name where the database is situated or the IP address of the server.

### Port

A TCP/IP port for connecting to the database server.

### Initial Database

The initial database to which user connects when making connection.

### User Name

User name for connecting to the database server.

### Password

Password for connecting to the server.

If your Internet Service Provider (ISP) does not provide direct access to its server, Secure Tunneling Protocol (SSH) / HTTP is another solution.

**Note:** For security reasons native remote direct connections to the PostgreSQL server are disabled. Therefore, you may not be able to use Navicat Premium or other similar PostgreSQL admin applications running on your computer to connect to the remote server. For more details, refer to next paragraph on Server Administration.

### For Server Administration:

By default, PostgreSQL only allows connections from the local machine using TCP/IP connections. Other machines will not be able to connect unless you modify *listen\_addresses* in the *postgresql.conf* file, enable host-based authentication by modifying the

`$PGDATA/pg_hba.conf` file, and restart the server. For more information: [Client Authentication](#)

## General Settings for SQLite

The following instruction guides you through the process of creating a new connection. To successfully establish a new connection to local/remote SQLite - no matter via HTTP, set the connection properties in the corresponding boxes: Connection name, Type and Database Name.

### Connection Name

A friendly name to best describe your connection.

### Type

Specify the type of database.

#### Existing Database File

Connect an existing database in the **Database File**.

#### New SQLite 3

Create a new SQLite 3 database in the **Database File**.

#### New SQLite 2

Create a new SQLite 2 database in the **Database File**.

### Database File

Specify the initial database file. If the HTTP Tunnel is enabled, you need to enter an absolute file path of the database file in your webserver.

## General Settings for SQL Server

The following instruction guides you through the process of creating a new connection. To successfully establish a new connection to local/remote SQL Server - no matter via SSH, set the connection properties in the corresponding boxes: Connection name, Host name, and Authentication Type.

### Connection Name

A friendly name to best describe your connection.

### Host Name/IP Address

A host name where the database is situated or the IP address of the server.

### Authentication

SQL Server uses two ways to validate connections to SQL Server databases: SQL Server Authentication and Windows Authentication.

#### SQL Server Authentication

SQL Server Authentication uses login records to validate the connection. Users must provide their login username and password every time that they connect.

##### User Name

User name for connecting to the database server.

##### Password

Password for connecting to the server.

#### Windows Authentication

When a user connects through a Windows user account, SQL Server validates the account name and password using the Windows principal token in the operating system. This means that the user identity is confirmed by Windows. SQL Server does not ask for the password, and does not perform the identity validation.

If your Internet Service Provider (ISP) does not provide direct access to its server, Secure Tunneling Protocol (SSH) is another solution.



## **SSH Settings (Available only for MySQL, Oracle, PostgreSQL and SQL Server and supports SSH2 Protocol only)**

**Secure SHell (SSH)** is a program to log in into another computer over a network, execute commands on a remote server, and move files from one machine to another. It provides strong authentication and secure encrypted communications between two hosts, known as **SSH Port Forwarding (Tunneling)**, over an insecure network. Typically, it is employed as an encrypted version of Telnet.

In a Telnet session, all communications, including username and password, are transmitted in plain-text, allowing anyone to listen-in on your session and steal passwords and other information. Such sessions are also susceptible to session hijacking, where a malicious user takes over your session once you have authenticated. SSH serves to prevent such vulnerabilities and allows you to access a remote server's shell without compromising security.

- [Benefit of SSH Tunneling.](#)

To ensure that the incoming connection request is from you, SSH can use a password, or public/private key pair (also called public key) authentication mechanism.

- [Password Authentication.](#)
- [Public Key Authentication.](#)

**Note:** Please make sure that the parameter - "AllowTcpForwarding" in the Linux Server must be set to value "yes", otherwise, the SSH port forwarding will be disabled. To look for the path: [/etc/ssh/sshd\\_config](#) .By default, the SSH port forwarding should be enabled. Please double check the value settings.

**\*\*** Even the server support SSH tunnel, however, if the port forwarding being disabled, Navicat cannot connect via SSH Port 22.

## Benefit of SSH Tunneling

SSH has a wonderful feature called SSH Port Forwarding, sometimes called SSH Tunneling, which allows you to establish a secure SSH session and then tunnel arbitrary TCP connections through it. Tunnels can be created at any time, with almost no effort and no programming, which makes them very appealing. SSH Port Forwarding can be used for secure communications in a myriad of different ways.

Many Hosting Companies that provide server hosting will block access to the Server from outside the hosting company's network, and only grant access to users connecting from localhost.


There are several benefits to using SSH:

- Connection to a server from behind a firewall when the server port is blocked.
- Automatic authentication of users, no passwords sent in plain text to prevent the stealing of passwords.
- Multiple strong authentication methods that prevent such security threats as spoofing identity.
- Encryption and compression of data for security and speed.
- Secure file transfer.

## Password Authentication

Using this mode, SSH is almost identical to the program telnet. When you make a connection, you are asked for your password. You type it in and you are either logged in or denied. Your password is first encrypted and then sent over the network and then decrypted at the remote host. This is the mode that most users will be encouraged to use, as it requires no additional setup or configuration.

The following instruction guides you through the process of configuring a SSH connection using Password Authentication. To successfully establish a SSH connection, set the SSH connection properties in the corresponding boxes: Host name/IP address, Port number, User name, Authentication Method and Password.

1. Click  or choose File ->  **New Connection** to set up the Connection Properties.
2. Select the SSH tab and enable **Use SSH Tunnel**.
3. Fill in the required information:

### Host Name/IP Address

A host where SSH server is activated.

### Port

A port where SSH server is activated, by default it is 22.

### User Name

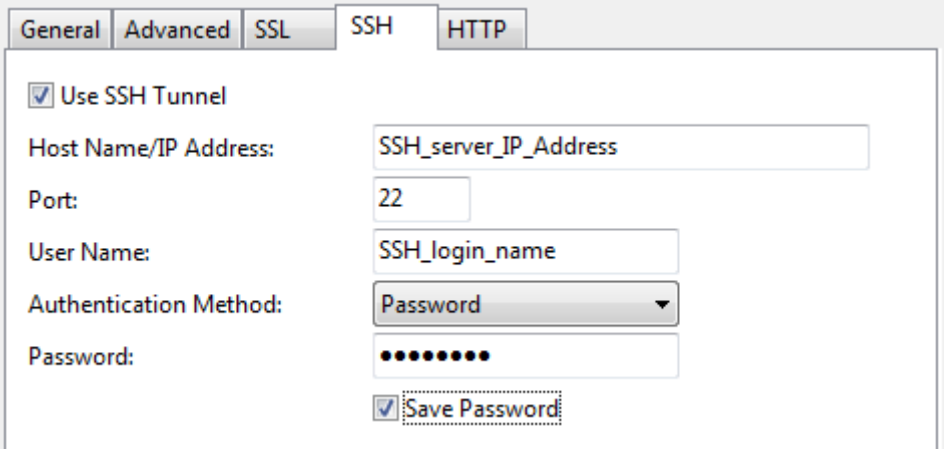
A user on Linux machine. (It is a Linux user. It is not a user of Database Server.)

### Authentication Method

Choose between **Password Authentication** and Public Key Authentication

### Password

It is a Linux user password.



General Advanced SSL **SSH** HTTP

☒ Use SSH Tunnel

Host Name/IP Address: SSH\_server\_IP\_Address

Port: 22

User Name: SSH\_login\_name

Authentication Method: Password ▼

Password: ●●●●●●●●

☒ Save Password

4. Navicat host name at the General Settings page should be set relatively to the SSH server which provided by your database hosting company.

## Public Key Authentication

Public-key Authentication is based on the use of digital signatures and provides the best authentication security.

For Public Key Authentication to work four things are needed:

- the remote server(s) you are connecting must have your public key.
- the local client you are connecting from must have your private key.
- the remote server must be configured to allow you to login using your public key.
- the local client must be configured to use your private key while logging into remote server.

The following instruction guides you through the process of configuring a SSH connection using Public Key Authentication. To successfully establish a SSH connection , set the SSH connection properties in the corresponding boxes: Host name/IP address, Port number, User name, Authentication Method, Private Key and Passphrase.

1. Click  or choose File ->  **New Connection** to set up the Connection Properties.
2. Select the SSH tab and enable **Use SSH Tunnel**.
3. Fill in the required information:

### Host Name/IP Address

A host where SSH server is activated.

### Port

A port where SSH server is activated, by default it is 22.

### User Name

A user on Linux machine. (It is a Linux user. It is not a user of Database Server.)

### Authentication Method

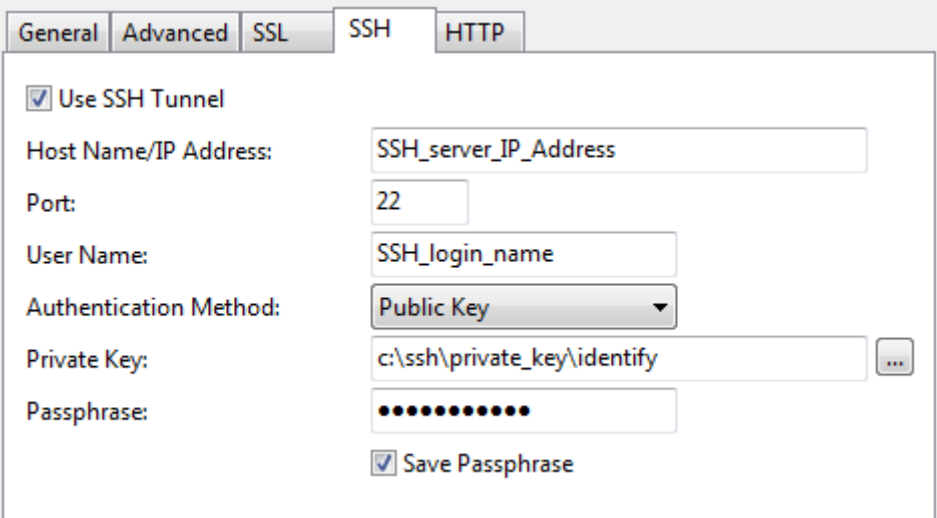
Choose between Password Authentication and **Public Key Authentication**

### Private Key

It is used together with your public key. The private key should be readable only by you.

### Passphrase

A passphrase is exactly like a password, except that it applies to the keys you are generating and not an account. The passphrase be any length under 1024 characters.



The screenshot shows the 'SSH' tab of the Navicat settings dialog. It includes fields for 'Host Name/IP Address' (SSH\_server\_IP\_Address), 'Port' (22), 'User Name' (SSH\_login\_name), 'Authentication Method' (Public Key), 'Private Key' (c:\ssh\private\_key\identify), and 'Passphrase' (masked with dots). There are checkboxes for 'Use SSH Tunnel' and 'Save Passphrase'.

Field	Value
Use SSH Tunnel	<input checked="" type="checkbox"/>
Host Name/IP Address:	SSH_server_IP_Address
Port:	22
User Name:	SSH_login_name
Authentication Method:	Public Key
Private Key:	c:\ssh\private_key\identify
Passphrase:	.....
Save Passphrase	<input checked="" type="checkbox"/>

4. Navicat host name at the General Settings page should be set relatively to the SSH server which provided by your database hosting company.

## HTTP Settings (Available only for MySQL, PostgreSQL and SQLite)

HTTP Tunneling is a method for connecting to a server that uses the same protocol (http://) and the same port (port 80) as a webserver does. It is used while your ISPs do not allow direct connections, but allows establishing HTTP connections.

Steps of setting up HTTP Connection:



### 1. Uploading the Tunneling Script

To use this connection method, first thing you need to do is to upload the tunneling script to the webserver where your server is located.

**Note:** `ntunnel_mysql.php`, `ntunnel_pgsql.php` or `ntunnel_sqlite.php` is available in the Navicat installation folder.

### 2. Setting up HTTP Tunnel

The following instruction guides you through the process of configuring a HTTP connection.

- i. Click  or choose File ->  **New Connection** to set up the Connection Properties.
- ii. Select the HTTP tab and enable **Use HTTP Tunnel**.
- iii. Enter URL of the tunneling script, e.g.  
`http://www.navicat.com/ntunnel_mysql.php` .
- iv. If your server installed ModSecurity, you can check the **Encode outgoing query with base64** option.
- v. If the tunneling script is hosted in a password protected server or you have to access internet over a proxy server, you can provide the required authentication details in **Authentication** or **Proxy** tab..
- vi. Navicat host name at the General settings page should be set relatively to the HTTP server which provided by your database hosting company.

**Note:** HTTP Tunnel and SSH Tunnel cannot function simultaneously. The SSH Tunnel is disabled when you select the HTTP Tunnel and vice versa.

## SSL Settings (Available only for MySQL and PostgreSQL)

Secure Sockets Layer(SSL) is a protocol for transmitting private documents via the Internet. To get a secure connection, the first thing you need to do is to install OpenSSL Library and download Database Source.

Steps of setting up SSL Connection and Navicat:

1. [Installation of OpenSSL and MySQL/PostgreSQL.](#)
2. [Setting up SSL Certificate for MySQL/PostgreSQL.](#)
3. [Setting up Client Certificate for Navicat.](#)

**Note:** Support from PostgreSQL 8.4 or later.



## Installation of OpenSSL and MySQL/PostgreSQL

### Installing OpenSSL

1. Download OpenSSL - <http://www.openssl.org>
2. Linux command : [`zcat 0.96l.tar.gz | tar xvf -`]
3. Linux command : [`./config`]
4. Linux command : [`make`]
5. Linux command : [`make install`]

### Installing MySQL

1. Download MySQL - <http://www.mysql.com>
2. Linux command : [`./configure --with -vio --with -openssl`]
3. Linux command : [`make`]
4. Linux command : [`make install`]

**Note:** Please ensure if MySQL Server supports OpenSSL using query statement:  
[SHOW VARIABLES LIKE 'have\_openssl';] - Returns value = YES

### Installing PostgreSQL

1. Download PostgreSQL - <http://www.postgresql.org>
2. Linux command : [`./configure --with-openssl`]
3. Linux command : [`gmake`]
4. Linux command : [`gmake install`]

**Note:** Please ensure if PostgreSQL Server supports OpenSSL using query statement:  
[SHOW ssl;] - Returns value = ON

## Setting up SSL Certificate for MySQL/PostgreSQL

To create server/client side Certificate, login to the Linux Server as root and employ the Shell Command below:

### MySQL

1. `DIR=`pwd`/openssl`
2. `PRIV=$DIR/private`
3. `mkdir $DIR $PRIV $DIR/newcerts`
4. `cp /usr/share/ssl/openssl.cnf $DIR`
5. `replace ./demoCA $DIR -- $DIR/openssl.cnf`
6. Generation of Certificate Authority(CA)

```
/usr/local/ssl/bin/openssl req -new -x509 -keyout $PRIV/cakey.pem -out  
$DIR/cacert.pem -config $DIR/openssl.cnf
```

**Note:** If "PEM" is required, please enter different "PEM pass" via steps below.

7. Create server request and key

```
/usr/local/ssl/bin/openssl req -new -keyout $DIR/server-key.pem -out  
$DIR/server-req.pem -days 3600 -config $DIR/openssl.cnf
```

8. Remove the passphrase from the key (optional)

```
/usr/local/ssl/bin/openssl rsa -in $DIR/server-key.pem -out $DIR/server-key.pem
```

9. Sign server cert

```
/usr/local/ssl/bin/openssl ca -policy policy_anything -out $DIR/server-cert.pem  
-config $DIR/openssl.cnf -infiles $DIR/server-req.pem
```

10. Create client request and key

```
/usr/local/ssl/bin/openssl req -new -keyout $DIR/client-key.pem -out  
$DIR/client-req.pem -days 3600 -config $DIR/openssl.cnf
```

11. Remove a passphrase from the key (optional)

```
/usr/local/ssl/bin/openssl rsa -in $DIR/client-key.pem -out $DIR/client-key.pem
```

12. Sign client cert

```
/usr/local/ssl/bin/openssl ca -policy policy_anything -out $DIR/client-cert.pem  
-config $DIR/openssl.cnf -infiles $DIR/client-req.pem
```

13. Create a **my.cnf** file for testing the Certificates. Store it either in **/etc** or MySQL data directory (typically **/usr/local/var** for source installation)

**my.cnf** example content:

```
[client]  
ssl-ca=$DIR/cacert.pem  
ssl-cert=$DIR/client-cert.pem  
ssl-key=$DIR/client-key.pem  
[mysqld]  
ssl-ca=$DIR/cacert.pem  
ssl-cert=$DIR/server-cert.pem  
ssl-key=$DIR/server-key.pem
```

14. To start MySQL daemon:

```
/usr/local/libexec/mysqld -u mysql &
```

or

```
/usr/local/sbin/mysqld -u &
```

## PostgreSQL

1. To create a quick self-signed certificate for the server, use the following OpenSSL command:

```
openssl req -new -text -out server.reqm
```

2. Fill out the information that openssl asks for. Make sure you enter the local host name as "Common Name"; the challenge password can be left blank. The program will generate a key that is passphrase protected; it will not accept a passphrase that is less than four characters long. To remove the passphrase (as you must if you want automatic start-up of the server), run the commands:

```
openssl rsa -in privkey.pem -out server.key  
rm privkey.pem
```

3. Enter the old passphrase to unlock the existing key. Now do:

```
openssl req -x509 -in server.req -text -key server.key -out server.crt
```



4. to turn the certificate into a self-signed certificate and to copy the key and certificate to where the server will look for them. Finally do:

```
chmod og-rwx server.key
```


## Setting up Client Certificate for Navicat

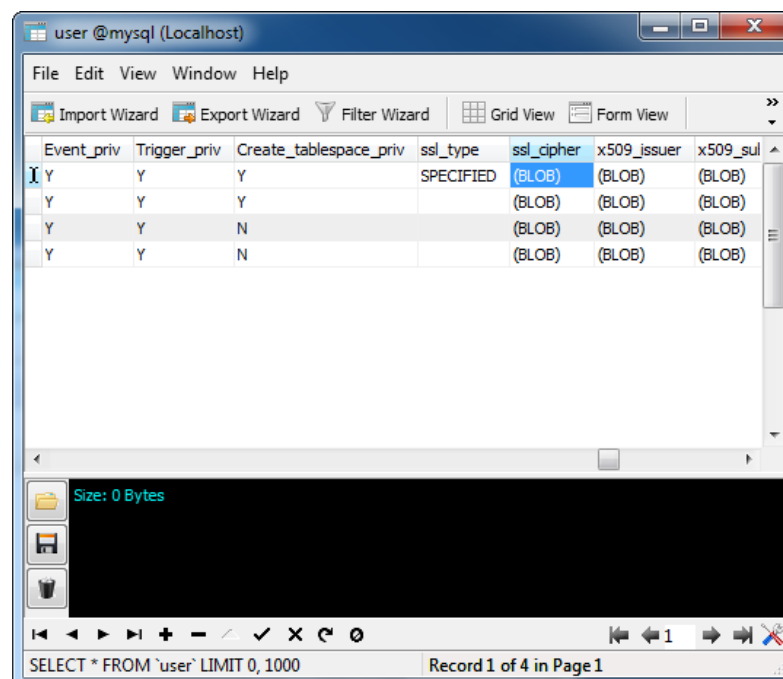
The following instruction guides you through the process of configuring a connection between Navicat and MySQL/PostgreSQL Server using SSL. To successfully establish a SSL connection, please complete Step 1 and Step 2, and set the connection properties in the corresponding boxes.

### MySQL



1. Click  or choose File ->  **New Connection** to set up the Connection Properties.
2. Select the SSL tab and enable **Use SSL**.
3. To provide authentication details, fill in the required information:

**Client Key**, **Client Certificate** and **CA Certificate** are usually stored in your Server - `/usr/local/openssl`. Please copy them from the remote server to local computer. **Specified Cipher** (optional) is only required while **ssl\_type** field has been set to **"SPECIFIED"** - [ssl\_type can be found in a system database called "mysql" -> table called "user"]. Example of Specified Cipher is "EDH-RSA-DES-CBC3-SHA" which can be filled in either through the Connection Properties shown above or the "mysql" database -> "user" table -> "ssl\_cipher" blob field shown below.

**Note:** You are allowed to store your Specified Cipher into a text file in order to load  into the "ssl\_cipher" blob field.



## PostgreSQL

1. Click  or choose File ->  **New Connection** to set up the Connection Properties.
2. Select the SSL tab and enable **Use SSL**.
3. Select the **SSL Mode**.
  - require** - only try an SSL connection.
  - verify-ca** - only try an SSL connection, and verify that the server certificate is issued by a trusted CA.
  - verify-full** - only try an SSL connection, verify that the server certificate is issued by a trusted CA and that the server hostname matches that in the certificate.
4. To provide authentication details, enable **Use Authentication** and fill in the required information:

**Client Key**, **Client Certificate** and **CA Certificate** are usually stored in your Server - `/usr/local/openssl`. Please copy them from the remote server to local computer.

**Certificate Revocation List** specifies the file path of the SSL certificate revocation list (CRL).

For PostgreSQL server, OpenSSL supports a wide range of ciphers and authentication algorithms, of varying strength. While a list of ciphers can be specified in the OpenSSL configuration file, you can specify ciphers specifically for use by the database server by modifying `ssl_ciphers` in `postgresql.conf`.

## Advanced Settings

Customize connection options according to your needs. The detailed description is given below:

### Settings Save Path

When a new connection being established, Navicat will create a subfolder under the Settings Save Path. Most files are stored within this subfolder:

Navicat Objects	Server Type	File Extensions
Query	All	.sql
Export Query Result Profile	MySQL	.npeq
	Oracle	.nopeq
	PostgreSQL	.nppeq
	SQLite	.nlpeq
	SQL Server	.nmpeq
Export View Result Profile	MySQL	.npev
	Oracle	.nopev
	PostgreSQL	.nppev
	SQLite	.nlpev
	SQL Server	.nmpev
Backup	MySQL, PostgreSQL and SQLite	compressed (.psc), uncompressed (.psb)
Backup Profile	MySQL	.npb
	PostgreSQL	.nppb
	SQLite	.nlpb
Report	All	.rtm
Import Wizard Profile	MySQL	.npi
	Oracle	.nopi
	PostgreSQL	.nppi
	SQLite	.nlpi
	SQL Server	.nmpi

Export Wizard Profile	MySQL	.npe
	Oracle	.nope
	PostgreSQL	.nppe
	SQLite	.nlpe
	SQL Server	.nmpe
Export Materialized View Profile	Oracle	.nopem
ER Diagram File	All	.ned

Other files are located in the **profiles** directory. To look for the path, choose Tools -> Options -> Miscellaneous -> Profiles Save Path.

Other Profiles	Server Type	File Extensions
Data Transfer	MySQL	.npt
	Oracle	.nopt
	PostgreSQL	.nppt
	SQLite	.nlpt
	SQL Server	.nmpt
	Premium (Cross Server)	.napt
Data Synchronization	MySQL	.npd
	Oracle	.nopd
	PostgreSQL	.nppd
	SQLite	.nlpd
	SQL Server	.nmpd
Structure Synchronization	MySQL	.nps
	Oracle	.nops
	PostgreSQL	.npps
	SQL Server	.nmps
Batch Job	Premium (Cross Server)	.napj
Model File	All	.ndm
Virtual Grouping	All	vgroup.xml - stores how the objects are categorized.

**Hint:** All your connection settings are stored in registry.

See also:

Log Files



## MySQL

### Encoding

Choose a codepage to communicate with MySQL Server while MySQL character set not being employed.

#### ☒ **Keepalive Interval (sec)**

This option allows you to keep the connection with the server alive by pinging it. You can set the period between pings in the edit field.

#### ☒ **Use MySQL character set**

This option should be enabled if employing MySQL 4.1 or above.

#### ☒ **Use Compression**

This option allows you to use compression protocol. It is used if both client and server support zlib compression, and the client requests compression.

#### ☒ **Auto Connect**

With this option on, Navicat automatically open connection with the registered database at application startup.

#### ☒ **Use Named Pipe, Socket**

With this option on, Navicat uses socket file for localhost connection.

## Oracle

### Role

Indicate that the database user is connecting with either the **Default**, **SYSOPER** or **SYSDBA** system privilege.

#### ☒ **Keepalive Interval (sec)**

This option allows you to keep the connection with the server alive by pinging it. You can set the period between pings in the edit field.

#### ☒ **OS Authentication**

With this option on, Oracle Database uses Windows user login credentials to authenticate database users.

#### ☒ **Auto Connect**

With this option on, Navicat automatically opens connection with the registered database at application startup.

## PostgreSQL

### ☒ **Keepalive Interval (sec)**

This option allows you to keep the connection with the server alive by pinging it. You can set the period between pings in the edit field

### ☒ **Auto Connect**

With this option on, Navicat automatically opens connection with the registered database at application startup.

## SQLite

### ☒ **Auto Connect**

With this option on, Navicat automatically opens connection with the registered database at application startup.

### ☒ **Encrypted**

Enable this option and provide **Password** when connecting to an encrypted SQLite database.

### **Attached Database**

To attach or detach databases in the connection.

## SQL Server

### **Initial Database**

The initial database to which user connects when making connection.

### ☒ **Keepalive Interval (sec)**

This option allows you to keep the connection with the server alive by pinging it. You can set the period between pings in the edit field.

### ☒ **Use Encryption**

This option allows you to use encryption.

### ☒ **Auto Connect**


With this option on, Navicat automatically opens connection with the registered database at application startup.

## Setting Advanced Database Properties (Available only for MySQL and PostgreSQL)


Set the advanced database properties, which are not obligatory. To start working with advanced database settings, check the **Use Advanced Connections**. The detailed description is given below:

### Show Selected Databases

To show the selected databases in the **close** state in the navigation pane

- Click the preferable databases in the Databases list box, the check box will show as 

To show the selected databases in the **open** state in the navigation pane

- Double-click the preferable databases in the Databases list box, the check box will show as 

### Add Hidden Database

To add a hidden database

- Click **Add DB to List** button.
- Enter the database name.
- Select the newly added database in the Databases list box.
- Enter **User Name** and **Password** which provide by your ISP.



### Remove Database


To remove a database

- Select the database to remove in the Databases list box.
- Click **Remove DB from List** button.

**Note:** The database will be just removed from the Databases list box, it will still exist in the Server.

## Working with Databases or Schemas

After you have created your connections, your databases/schemas appear in the navigation pane on the left. If the **Show objects in connection tree** option is checked at the Options window, all database/schema objects are also displayed in the pane. To connect to a database/schema, simply double-click it in the pane. If connection succeeds, the database/schema node turns into  or  and expands showing the tree of its objects.



For Oracle server, when you create a user account, you are also implicitly creating a schema for that user. A schema is a logical container for the database objects (such as tables, views, triggers, and so on) that the user creates. The schema name is the same as the user name, and can be used to unambiguously refer to objects owned by the user. Other user schemas are showed under  **Schemas**.

## MySQL Database Management

Navicat provides all the tools you need to manage and navigate databases. Note that to start working with databases in Navicat you are to establish the connection.

### Create Database

To create a database

- Double-click the connection to open in the navigation pane.
- Right-click the opened connection and choose  **New Database...**  
or
- Right-click any existing database and choose  **New Database...**
- Fill in the required information:

#### Enter database name

Set the name for a new database.

#### Character set


The Character set specifies the default database character set.

#### Collation

The Collation specifies the default database collation.

### Delete Database


To delete a database

- Right-click the database in the navigation pane and choose  **Delete Database.**
- Confirm deleting in the dialog window.


**Note:** This operation is irreversible.

### Open Database

To open a hidden database


- Double-click the connection to open in the navigation pane.
- Right-click the opened connection and choose  **Open Database.**
- Enter the database name.

To open a database which shows in the navigation pane

- Double-click the database to open in the navigation pane.  
or
- Right-click the database and choose  **Open Database**.


## Close Database

To close a database

- Right-click the database in the navigation pane and choose  **Close Database**.

## Edit Database

To edit a database

- Right-click the database in the navigation pane and choose  **Database Properties....**

Please notice that MySQL does not support renaming database through its interface at this moment.

Access the directory in which databases being stored. By default, all databases store within a directory called **data** under MySQL Installation folder. For example: **C:\mysql5\data**.


**Note:** You must stop MySQL before you can rename the database.

## Oracle Schema Management

To start working with schemas in Navicat you are to establish the connection.


### Open Schema

To open a schema which shows in the navigation pane

- Double-click the schema to open in the navigation pane.  
or
- Right-click the schema and choose  **Open Schema**.

### Close Schema

To close a schema



- Right-click the schema in the navigation pane and choose  **Close Schema**.

## PostgreSQL Database Management

To start working with databases in Navicat you are to establish the connection.


### Create Database

To create a new database

- Double-click the connection to open in the navigation pane.
- Right-click the opened connection and choose  **New Database...**  
or
- Right-click any existing database and choose  **New Database...**
- Edit database properties on the appropriate tabs of the Database Designer.


### Edit Database

To edit the existing database(manage its general etc)

- Right-click the database in the navigation pane and choose  **Database Properties...**
- Edit database properties on the appropriate tabs of the Database Designer.


### Delete Database

To delete a database

- Right-click the database in the navigation pane and choose  **Delete Database.**
- Confirm deleting in the dialog window.


### Open Database

To open a hidden database

- Double-click the connection to open in the navigation pane.
- Right-click the opened connection and choose  **Open Database.**
- Enter the database name.




To open a database which shows in the navigation pane

- Double-click the database to open in the navigation pane.  
or
- Right-click the database and choose  **Open Database**.

## **Close Database**

To close a database

- Right-click the database in the navigation pane and choose  **Close Database**.

## PostgreSQL Database Designer

**Database Designer** is the basic Navicat tool for working with PostgreSQL database. It allows you to create new database and edit the existing database properties.

- [Editing Database General](#)
- Editing Database Comment

## Editing PostgreSQL Database General

To create a database, you must have the **Can create database** (usecreatedb) right. Refer to Role Editor or User Editor on how to set user properties.

### Database Name

Define the name of the database.

### Encoding

Define the encoding for the database. If omitted, the default is the encoding of the template database.

### Owner

Define the owner for the database. If omitted, defaults to the user executing the command. Only superusers may create database owned by users other than themselves.

### Template

Create the database from a template database.

**Note:** It is essential that the source database be idle (no data-altering transactions in progress) for the duration of the copying operation. CREATE DATABASE will check that no session (other than itself) is connected to the source database at the start of the operation, but this does not guarantee that changes cannot be made while the copy proceeds, which would result in an inconsistent copied database. Therefore, it is recommended that databases used as templates be treated as read-only.

### Tablespace



Define the tablespace for the database. If omitted, defaults to pg\_default.

## SQLite Database Management

Navicat provides all the tools you need to manage and navigate databases. Note that to start working with databases in Navicat you are to establish the connection.

### Attach Database

To attach a database

- Double-click the connection to open in the navigation pane.
- Right-click the opened connection and choose  **Attach Database**.  
or
- Right-click any existing database and choose  **Attach Database**.
- Fill in the required information:

#### Database File

Set the file path for a database.

#### Database Name


Enter the database name which displays in Navicat.

#### ☒ Encrypted

Enable this option and provide **Password** when connecting to an encrypted SQLite database.


### Open Database

To open a database which shows in the navigation pane

- Double-click the database to open in the navigation pane.  
or
- Right-click the database and choose  **Open Database**.

### Detach Database

To detach a database

- Right-click the database in the navigation pane and choose  **Detach Database**.

## Encrypt Database

To encrypt a database

- Right-click the database in the navigation pane and choose **Encrypt Database**.
- Enter the password.


## Decrypt Database

To decrypt a database

- Right-click the database in the navigation pane and choose **Decrypt Database**.
- Confirm decrypting in the dialog window.

## Close Database

To close a database



- Right-click the database in the navigation pane and choose  **Close Database**.

## SQL Server Database Management

To start working with databases in Navicat you are to establish the connection.


### Create Database

To create a new database

- Double-click the connection to open in the navigation pane.
- Right-click the opened connection and choose  **New Database....**  
or
- Right-click any existing database and choose  **New Database....**
- Edit database properties on the appropriate tabs of the Database Designer.


### Edit Database

To edit the existing database(manage its general etc)

- Right-click the database in the navigation pane and choose  **Database Properties....**
- Edit database properties on the appropriate tabs of the Database Designer.


### Delete Database

To delete a database

- Right-click the database in the navigation pane and choose  **Delete Database.**
- Confirm deleting in the dialog window.


### Open Database

To open a database

- Double-click the database to open in the navigation pane.  
or
- Right-click the database and choose  **Open Database.**

### Close Database

To close a database

- Right-click the database in the navigation pane and choose  **Close Database**.

## SQL Server Database Designer

**Database Designer** is the basic Navicat tool for working with database. It allows you to create new database and edit the existing database properties.

- [Editing Database General](#)
- [Editing Database Filegroups](#)
- [Editing Database Files](#)
- [Editing Advanced Database Properties](#)
- Editing Database Comment (SQL Azure does not support)
- Database SQL Preview

## Editing SQL Server Database General

### Options for SQL Server

#### Database Name

Define the name of the database.

#### Owner

Choose the owner of the database.

#### Collation

Choose the default collation for the database. Collation name can be either a Windows collation name or a SQL collation name. If not specified, the database is assigned the default collation of the instance of SQL Server. A collation name cannot be specified on a database snapshot.

#### Recovery Model

Control database recovery options and disk I/O error checking.

##### **FULL**

Provide full recovery after media failure by using transaction log backups. If a data file is damaged, media recovery can restore all committed transactions.

##### **BULK\_LOGGED**

Provide recovery after media failure by combining the best performance and least amount of log-space use for certain large-scale or bulk operations.

##### **SIMPLE**

A simple backup strategy that uses minimal log space is provided. Log space can be automatically reused when it is no longer required for server failure recovery.

#### Compatibility Level

Choose the version of SQL Server with which the database is to be made compatible.

### Options for SQL Azure

#### Database Name

Define the name of the database.



## **Edition**

Choose the edition of the database: web or business.

## **Max Size**

Choose the maximum size of the database.

## Editing SQL Server Database Filegroups

SQL Azure does not support this tab.

### **Filegroups**

Add or delete a filegroup. PRIMARY filegroup cannot be deleted.

### **FILESTREAM Filegroups**

Add or delete a FILESTREAM filegroup.

**Note:** Support from SQL Server 2008 or later.

## Editing SQL Server Database Files

SQL Azure does not support this tab.

### Database Files

#### Name

Specify the logical name for the file.

#### Type

Choose the file type.

#### Filegroup

Choose the filegroup.

### File Directory

The path used by the operating system when you create the file.

### File Name

The file name used by the operating system when you create the file.

### Size

Specify the size of the file.

#### ☒ Allow Auto Growth

Check this option if you want to allow automatic growth.

### Growth

Specify the automatic growth increment of the file.

### Max Size

Specify the maximum size to which the file can grow.

#### ☒ Unlimited

Specify that the file grows until the disk is full. In SQL Server, a log file specified with unlimited growth has a maximum size of 2 TB, and a data file has a maximum size of 16 TB.

## Editing Advanced SQL Server Database Properties

SQL Azure does not support this tab.

### State

#### ☒ **Database Read Only**

If this option is on, users can read data from the database but not modify it.

### **Database State**

Choose the state of the database.

#### **OFFLINE**

The database is closed, shut down cleanly, and marked offline. The database cannot be modified while it is offline.

#### **ONLINE**

The database is open and available for use.

#### **EMERGENCY**

The database is marked READ\_ONLY, logging is disabled, and access is limited to members of the sysadmin fixed server role. EMERGENCY is primarily used for troubleshooting purposes.

### **Restrict Access**

Control user access to the database.

#### **SINGLE\_USER**

Specifies that only one user at a time can access the database.

#### **RESTRICTED\_USER**

RESTRICTED\_USER allows for only members of the db\_owner fixed database role and dbcreator and sysadmin fixed server roles to connect to the database, but does not limit their number.

#### **MULTI\_USER**

All users that have the appropriate permissions to connect to the database are allowed.

## ☒ **Encryption Enabled**

Checks this option if you want to encrypt the database.

**Note:** Support from SQL Server 2008 or later.

## **SQL**

### ☒ **ANSI Null Default**

Checks this option if you want to determines the default value as NULL.

### ☒ **ANSI Nulls Enabled**

If this option is on, all comparisons to a null value evaluate to UNKNOWN.

### ☒ **ANSI Padding Enabled**

If this option is on, strings are padded to the same length before conversion or inserting to a varchar or nvarchar data type.

### ☒ **ANSI Warnings Enabled**

If this option is on, errors or warnings are issued when conditions such as divide-by-zero occur or null values appear in aggregate functions.

### ☒ **Arithmetic Abort Enabled**

If this option is on, a query is ended when an overflow or divide-by-zero error occurs during query execution.

### ☒ **Concatenate Null Yields Null**

If this option is on, the result of a concatenation operation is NULL when either operand is NULL.

### ☒ **Numeric Round Abort**

If this option is on, an error is generated when loss of precision occurs in an expression.

### ☒ **Quoted Identifiers Enabled**

If this option is on, double quotation marks can be used to enclose delimited identifiers.

### ☒ **Recursive Triggers Enabled**

If this option is on, Recursive firing of AFTER triggers is allowed.

## Cursor

### ☒ **Close Cursor On Commit Enabled**

If this option is on, any cursors open when a transaction is committed or rolled back are closed.

## Default Cursor

### **LOCAL**

When LOCAL is specified and a cursor is not defined as GLOBAL when created, the scope of the cursor is local to the batch, stored procedure, or trigger in which the cursor was created. The cursor name is valid only within this scope. The cursor can be referenced by local cursor variables in the batch, stored procedure, or trigger, or a stored procedure OUTPUT parameter. The cursor is implicitly deallocated when the batch, stored procedure, or trigger ends, unless it was passed back in an OUTPUT parameter. If the cursor is passed back in an OUTPUT parameter, the cursor is deallocated when the last variable that references it is deallocated or goes out of scope.

### **GLOBAL**

When GLOBAL is specified, and a cursor is not defined as LOCAL when created, the scope of the cursor is global to the connection. The cursor name can be referenced in any stored procedure or batch executed by the connection.

## Automatic

### ☒ **Auto Close**

If this option is on, the database is shut down cleanly and its resources are freed after the last user exits.

### ☒ **Auto Create Statistics**

If this option is on, the query optimizer creates statistics on single columns in query predicates, as necessary, to improve query plans and query performance.

### ☒ **Auto Shrink**

If this option is on, the database files are candidates for periodic shrinking.

### ☒ **Auto Update Statistics**

Specify that the query optimizer updates statistics when they are used by a query and when they might be out-of-date.

## ☒ **Auto Update Statistics Asynchronously**

Specify that statistics updates for the `AUTO_UPDATE_STATISTICS` option are asynchronous. The query optimizer does not wait for statistics updates to complete before it compiles queries.

**Note:** Support from SQL Server 2005 or later.

## **Recovery**

### **Page Verify**

Discovers damaged database pages caused by disk I/O path errors. Disk I/O path errors can be the cause of database corruption problems and are generally caused by power failures or disk hardware failures that occur at the time the page is being written to disk.

#### **NONE**

Database page writes will not generate a `CHECKSUM` or `TORN_PAGE_DETECTION` value. SQL Server will not verify a checksum or torn page during a read even if a `CHECKSUM` or `TORN_PAGE_DETECTION` value is present in the page header.

#### **TORN\_PAGE\_DETECTION**

Save a specific 2-bit pattern for each 512-byte sector in the 8-kilobyte (KB) database page and stored in the database page header when the page is written to disk.

#### **CHECKSUM**

Calculate a checksum over the contents of the whole page and stores the value in the page header when a page is written to disk.

## **Service Broker**

**Note:** Support from SQL Server 2005 or later.

### ☒ **Broker Enabled**

Specify that Service Broker is enabled for the specified database. Message delivery is started, and the `is_broker_enabled` flag is set to true in the `sys.databases` catalog view. The database retains the existing Service Broker identifier.

### ☒ **Honor Broker Priority**

Send operations take into consideration the priority levels that are assigned to conversations. Messages from conversations that have high priority levels are sent before messages from conversations that are assigned low priority levels.

**Note:** Support from SQL Server 2008 or later.

## Change Tracking

**Note:** Support from SQL Server 2008 or later.

### ☒ **Change Tracking Enabled**

Enable change tracking for the database. When you enable change tracking, you can also set the AUTO CLEANUP and CHANGE RETENTION options.

### **Retention Period**

Specify the minimum period for keeping change tracking information in the database. Data is removed only when the AUTO\_CLEANUP value is ON.

### ☒ **Auto Clean Up**

Change tracking information is automatically removed after the specified retention period.

## Miscellaneous

**Note:** Support from SQL Server 2005 or later.

### ☒ **Cross Database Ownership Chaining Enabled**

If this option is on, database can be the source or target of a cross-database ownership chain.

### ☒ **Trustworthy**

If this option is on, database modules (for example, user-defined functions or stored procedures) that use an impersonation context can access resources outside the database.

### ☒ **Date Correlation Optimization Enabled**

SQL Server maintains correlation statistics between any two tables in the database that are linked by a FOREIGN KEY constraint and have datetime columns.

## Parameterization

### **SIMPLE**

Queries are parameterized based on the default behavior of the database.



## **FORCED**

SQL Server parameterizes all queries in the database.

## ☒ **VarDecimal Storage Enabled**

Indicate that decimal and numeric data types are stored by using the vardecimal storage format.

## **Working with Database or Schema Objects**

You are authorized to access your database or schema objects of the selected database through the nodes of the navigation pane. Depending on the server version you connected, the supported objects will appear in the tree. To open a particular object, double-click to open in the appropriate editor. Right-click the object to display the popup menu, which allows you to perform various operations over the selected object or database.

## Viewing Object Information

To view the object information, just simply select an object in the navigation pane/object pane and click **View** -> **Object Information** or right-click an object and choose **XXX Information** to open an **Object Information** in the object pane.

- **General**

Shows the object information as a grid.

- **DDL**

Shows the DDL statement of the object.

- **Using**

Shows the objects that the current object used.

**Note:** Available only for Oracle, PostgreSQL and SQL Server.

- **Used by**

Shows the current object used by whom.

**Note:** Available only for Oracle, PostgreSQL and SQL Server.

- **Objects**

Shows the objects in the tablespace.

**Note:** Available only for Oracle and PostgreSQL.

- **Preview**

Shows the sql statement in the query.

- **Member of**

Shows the roles that the user or the role assigned to.

**Note:** Available only for Oracle, PostgreSQL and SQL Server.

- **Members**

Shows the members of the role.

**Note:** Available only for Oracle, PostgreSQL and SQL Server.

## Database Object Management

Navicat provides powerful tools to manage MySQL, Oracle, PostgreSQL, SQLite and SQL Server database objects.

**Note:** before working with database objects in Navicat you should establish the connection first.

- [MySQL Database Object Management](#)
- [Oracle Database Object Management](#)
- [PostgreSQL Database Object Management](#)
- [SQLite Database Object Management](#)
- [SQL Server Database Object Management](#)


## MySQL Database Object Management

The following list contains the most common MySQL database objects supported by Navicat.

- [Tables](#)
- [Views](#)
- [Functions/Procedures](#)
- [Events](#)



## MySQL Tables


Relational databases use tables to store data. All operations on data are done on the tables themselves or produce another table as the result. A table is a set of rows and columns, and their intersections are fields. From a general perspective, columns within a table describe the name and type of data that will be found by row for that column's fields. Rows within a table represent records composed of fields that are described from left to right by their corresponding column's name and type. Each field in a row is implicitly correlated with each other field in that row.

Just simply click  to open an object pane for **Table**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit, open and delete the selected table.

### Create Table

To create a new table

- Select anywhere on the object pane.
- Click the  **New Table** from the object pane toolbar.  
or
- Right-click and select  **New Table** from the popup menu.
- Edit table properties and fields on the appropriate tabs of the Table Designer.

**Hint:** To create new table you can also right-click the Tables node of the navigation pane and select the  **New Table** from the popup menu.

To create a new table with the same properties as one of the existing tables has (using popup menu)

**Apply to:** current database {same connection}

- Select the table(s) for copying in the navigation pane/object pane.
- Right-click and select the **Duplicate Table** from the popup menu.
- The newly created table(s) will be named as "tablename\_**copy**".

To create a new table with the same properties as one of the existing tables has (using drag and drop method)

**Apply to:** current database {same connection}




- Select the table(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen table(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created table(s) will be named as "tablename\_**copy**"

**Apply to:** different database {same connection}

different database {different connection (same or cross server type)} (Data Transfer tool will be activated)

- Select the table(s) for copying in the object pane.
- Drag and drop the chosen table(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel

To create a new table with modification as one of the existing tables

- Select the table for modifying in the navigation pane/object pane.
- Right-click and select the  **Design Table** from the popup menu.  
or
- Click the  **Design Table** from the object pane toolbar.
- Modify table properties and fields on the appropriate tabs of the Table Designer.
- Click  **Save As**.

## Create Table Shortcut



To create a table shortcut

- Select the table for editing in the navigation pane/object pane.
- Right-click and select **Create Open Table Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.

**Note:** This option is used to provide a convenient way for you to open your table for entering data directly (Grid View/Form View) without activating the main Navicat.

## Edit Table

To edit the existing table (manage its fields, indexes, foreign keys and triggers etc)



- Select the table for editing in the navigation pane/object pane.
- Right-click and select the  **Design Table** from the popup menu.  
or
- Click the  **Design Table** from the object pane toolbar.
- Edit table properties and fields on the appropriate tabs of the Table Designer.


To change the name of the table

- Select the table for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.

## Open Table (manage table data)


To open a table

- Select the table for opening in the navigation pane/object pane.
- Right-click and select the  **Open Table** from the popup menu or simply double-click the table.  
or
- Click the  **Open Table** from the object pane toolbar.

**Note:** This option is only applied if you do wish Navicat loads all your images while opening the table. To open the graphical table with faster performance, use  **Open Table (Quick)** below.



To open a table with graphical fields

- Select the table for opening in the navigation pane/object pane.
- Right-click and select the  **Open Table (Quick)** from the popup menu.

**Note:** Faster performance for opening the graphical table, as BLOB fields (images) will not be loaded until you click on the cell.

## Empty Table

To empty a table

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Empty Table** from the popup menu.

**Note:** This option is only applied when you wish to clear all the existing records without resetting the auto-increment value. To reset the auto-increment value while emptying your table, use **Truncate Table** below.



## Truncate Table

To truncate a table

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Truncate Table** from the popup menu.

## Delete Table

To delete a table

- Select the table for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Table** from the popup menu.  
or
- Click the  **Delete Table** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Table Information

To achieve a table information

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## MySQL Table Designer

**Table Designer** is the basic Navicat tool for working with tables. It allows you to create, edit and drop table's fields, indexes, foreign keys, and much more.



- [Managing Table Fields](#)
- [Managing Table Indexes](#)
- [Managing Table Foreign Keys](#)
- [Managing Table Triggers](#)
- [Managing Table Options](#)
- Managing Table Comment
- Table SQL Preview

## MySQL Table Fields

Table fields are managed on the **Fields** tab of the Table Designer. Just simply click a field for editing. A right-click displays the popup menu or using field toolbar, allowing you to create new, insert, move and drop the selected field.

### Add Field

To add a field to the table



- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click and select the  **Add Field** from the popup menu or click the  **Add Field** from the toolbar.
- Edit field properties.

To add a new field with modification as one of the existing fields

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Select field.
- Right-click and select the **Duplicate Field** from the popup menu.
- Edit field properties.

### Insert Field

To insert a field above an existing field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Select field.
- Right-click and select the  **Insert Field** from the popup menu or click the  **Insert Field** from the toolbar.
- Define field properties in the empty row.





**Note:** Support from MySQL 3.22 or later.

## Edit Field

To edit the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Simply click on the field to edit.



To change the order of the table fields

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click on the field to move and select the  **Move Up**/  **Move Down** from the popup menu or click the  **Move Up**/  **Move Down** from the toolbar.


**Note:** Support from MySQL 4.0.1 or later.

## Delete Field

To delete the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click on the field to delete and select the  **Delete Field** from the popup menu or click the  **Delete Field** from the toolbar.
- Confirm deleting in the dialog window.

## Setting MySQL Table Field Properties

Name	Type	Length	Decimals	Allow Null	
CustNo	double	0	0	<input type="checkbox"/>	 1
Company	varchar	30	0	<input checked="" type="checkbox"/>	
Addr1	varchar	30	0	<input checked="" type="checkbox"/>	
Addr2	varchar	30	0	<input checked="" type="checkbox"/>	

### Name

The **Name** is a descriptive identifier for a field that can be up to 64 characters (letters or numbers) including spaces. The names should be descriptive enough that anyone can easily identify them when viewing or editing records. For example, LastName, FirstName, StreetAddress, or HomePhone.

Use the **Name** edit box to set the field name. Note that the name of the field must be unique among all the field names in the table.

### Type

After you name a field, you choose a data type for the data to be contained in the field. When you choose a field's data type, you are deciding:

- What kind of values to allow in the field. You cannot store text in field with the **Numeric** data type.
- How much storage space MySQL is to set aside for the data in that field.
- What types of operations can be performed on the values in that field.

The **Type** dropdown list defines the type of the field data. See [MySQL Data Types](#) for details.

### Length and Decimals

Use the **Length** edit box to define the length of the field and use **Decimals** edit box to define the number of digits after the decimal point (the scale) for Floating Point data type.

**Note:** Be careful when shortening the field length as losing data might be caused.

### ☒ Allow Null

Allow the NULL values for the field.

## **Primary Key**

A **Primary Key** is a single field or combination of fields that uniquely defines a record. None of the fields that are part of the primary key can contain a null value.

## Setting MySQL Table Other Field Properties

To set the default value for the field use the **Default** edit box.

**Note:** TEXT(tinytext, text, mediumtext and longtext) and BLOB(tinyblob, blob, mediumblob and longblob) data type cannot have **DEFAULT** values.

To set any optional text describing the current field use the **Comment** edit box.

**Note:** Apply to all data type.

To set other field properties for Text/Memo and BLOB (Binary Large Object) (not apply to binary/varbinary type)

### **Character set** (non-binary strings only)

A **character set** is a set of symbols and encodings. The **Character set** drop-down list defines the type of the character set for field.

### **Collation** (non-binary strings only)

A **collation** is a set of rules for comparing characters in a character set. The **Collation** drop-down list defines the type of the collation for field.

**Note:** MySQL chooses the column character set and collation in the following manner:

- If both CHARACTER SET X and COLLATE Y were specified, then character set X and collation Y are used.
- If CHARACTER SET X was specified without COLLATE, then character set X and its default collation are used.
- Otherwise, the table character set and collation are used.

### **Key Length**

The edit box will be enabled when Primary Key is set. KEY LENGTH (1 - 255).

### ☒ **Binary** (char and varchar only)

As of MySQL 4.1, values in CHAR and VARCHAR fields are sorted and compared according to the collation of the character set assigned to the field.



Before MySQL 4.1, sorting and comparison are based on the collation of the server character set; you can declare the field with the BINARY attribute to cause sorting and comparison to be based on the numeric values of the bytes in field values. BINARY does not affect how field values are stored or retrieved.

To set other field properties for Number/Currency and Floating Point (not apply to bit type)

☒ **Auto Increment** (Number/Currency only)

The AUTO INCREMENT attribute can be used to generate a unique identity for new rows. To start with the AUTO INCREMENT value other than 1, you can set that value in Options tab.

☒ **Unsigned**

UNSIGNED values can be used when you want to allow only non-negative numbers in a field and you need a bigger upper numeric range for the field.

As of MySQL 4.0.2, floating-point and fixed-point types also can be UNSIGNED. Unlike the integer types, the upper range of column values remains the same.

☒ **Zerofill**

The default padding of spaces is replaced with zeros. For example, for a field declared as INT(5) ZEROFILL, a value of 4 is retrieved as 00004; for a field declared as FLOAT(20,10) ZEROFILL, a value of 0.1 is retrieved as 000000000.1000000015.

**Note:** If you specify ZEROFILL for a numeric type, MySQL automatically adds the UNSIGNED attribute to the field.

To set other field properties for Date/Time

☒ **On Update Current\_Timestamp** (timestamp only)

As of 4.1.2, you have more flexibility in deciding which TIMESTAMP field automatically is initialized and updated to the current timestamp.

To set other field properties for Set/Enumerate

## Values

Use **Values** edit box to define the members of SET/ENUM. A SET field can have a maximum of 64 members. An ENUM field can have a maximum of 65,535 distinct values.



## MySQL Table Indexes

Indexes are organized versions of specific columns in your tables. MySQL uses indexes to facilitate quick retrieval of records. With indexes, MySQL can jump directly to the records you want. Without any indexes, MySQL has to read the entire data file to find the correct record(s).

Table indexes are managed on the **Indexes** tab of the Table Designer. Just simply click/double-click an index field for editing. A right-click displays the popup menu or using the index toolbar, allowing you to create new, edit and delete the selected index field.

### Add Index

To add a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Right-click and select the  **Add Index** from the popup menu or click the  **Add Index** from the toolbar.
- Edit index properties.



### Edit Index

To edit a table index


- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Just simply click/double-click on the index to edit.

### Delete Index


To delete a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Right-click on the index to delete and select the  **Delete Index** from the popup menu or click the  **Delete Index** from the toolbar.
- Confirm deleting in the dialog window.

## Setting MySQL Table Index Properties

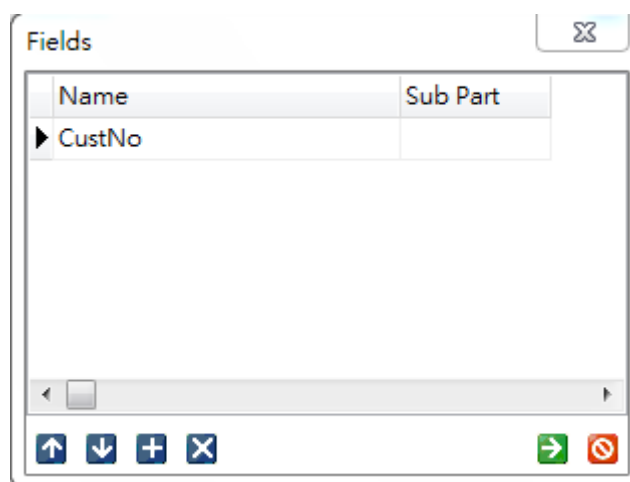
Name	Fields	Index Type	Index method
▶ CustNo	CustNo  	Normal	BTREE

Use the **Name** edit box to set the index name.

To include field(s) in the index, just simply double-click the **Fields** field or click  to open the editor for editing.

Select the field(s) from the list. To remove the fields from the index, uncheck them in the same way. You can also use the arrow buttons to change the index field(s) order. The **Sub Part** edit box(s) is used to set index KEY LENGTH (1 - 255).

**Note:** Some of data types do not allow indexing by several fields. For example: BLOB



The **Index Type** dropdown list defines the type of the table index.

### Normal

NORMAL indexes are the most basic indexes, and have no restraints such as uniqueness.

### Unique

UNIQUE indexes are the same as NORMAL indexes with one difference - all values of the indexed column(s) must only occur once.

### Full Text

FULL TEXT indexes are used by MySQL in full-text searches.

## **Index method**

Specify an index type when creating an index, BTREE or HASH.

## MySQL Table Foreign Keys



A foreign key is a field in a relational table that matches the primary key column of another table. The foreign key can be used to cross-reference tables.

Foreign Keys are managed on the **Foreign Keys** tab of the Table Designer. Just simply click/double-click a foreign key field for editing. A right-click displays the popup menu or using the foreign key toolbar, allowing you to create new, edit and delete the selected foreign key field.

**Note:** Foreign Key support from MySQL 3.23.44 or later.

### Add Foreign Key

To add a foreign key

- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Right-click and select the  **Add Foreign Key** from the popup menu or click the  **Add Foreign Key** from the toolbar.
- Edit foreign key properties.

**Note:** Both tables must be *InnoDB* type (or *solidDB* type if you have [solidDB for MySQL](#)). In the referencing table, there must be an index where the foreign key columns are listed as the first columns in the same order. Starting with MySQL 4.1.2, such an index will be created on the referencing table automatically if it does not exist.

### Edit Foreign Key



To edit a foreign key

- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Just simply click/double-click on the foreign key to edit.

**Note:** Support from MySQL 4.0.13 or later.


## Delete Foreign Key

To delete a foreign key

- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Right-click on the foreign key to delete and select the  **Delete Foreign Key** from the popup menu or click the  **Delete Foreign Key** from the toolbar.
- Confirm deleting in the dialog window.


**Note:** Support from MySQL 4.0.13 or later.

## Setting MySQL Table Foreign Key Properties

	Name	Fields	Reference Database	Reference Table	Reference Fields	On Delete	On Update
I	cust_no_fk	CustNo	report_sample	customer	CustNo 	RESTRICT	RESTRICT

Use the **Name** edit box to enter a name for the new key and then select a table field to include in the key from the **Fields** group.

Use the **Reference Database** and **Reference Table** dropdown lists to select a foreign database and table respectively.

To include field(s) to the key, just simply double-click the **Fields/Reference Fields** field or click  to open the editor(s) for editing.

The **On Delete** and **On Update** dropdown list define the type of the actions to be taken.

### CASCADE

Delete the corresponding foreign key, or update the corresponding foreign key to the new value of the primary key.

### SET NULL

Set all the columns of the corresponding foreign key to NULL.

### No ACTION

Does not change the foreign key.

### RESTRICT

Produce an error indicating that the deletion or update would create a foreign key constraint violation.

## MySQL Table Triggers



A trigger is a named database object that is associated with a table and that is activated when a particular event occurs for the table.

Triggers are managed on the **Triggers** tab of the Table Designer. Just simply click a trigger field for editing. A right-click displays the popup menu or using the trigger toolbar, allowing you to create new, edit and delete the selected trigger field.

**Note:** Trigger is supported from MySQL 5.0.2 or later.

### Add Trigger

To add a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Right-click and select the  **Add Trigger** from the popup menu or click the  **Add Trigger** from the toolbar.
- Edit trigger properties.



### Edit Trigger

To edit a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Just simply click on the trigger to edit.

### Delete Trigger

To delete a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Right-click on the trigger to delete and select the  **Delete Trigger** from the popup menu or click the  **Delete Trigger** from the toolbar.
- Confirm deleting in the dialog window.



## Setting MySQL Table Trigger Properties

Name	Fires	Insert	Update	Delete
► Capacity_Tri	After	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Use the **Name** edit box to set the trigger name.

Use the **Fires** dropdown list to define the trigger action time. It can be **Before** or **After** to indicate that the trigger activates before or after the statement that activated it.

### ☒ **Insert**

The trigger is activated whenever a new row is inserted into the table. For example, **INSERT**, **LOAD DATA**, and **REPLACE** statements.

### ☒ **Update**

The trigger is activated whenever a row is modified. For example, **UPDATE** statement.

### ☒ **Delete**

The trigger is activated whenever a row is deleted from the table. For example, **DELETE** and **REPLACE** statement. However, **DROP TABLE** and **TRUNCATE** statements on the table do not activate the trigger.

The **Definition** tab defines the statement to execute when the trigger activates. To include your statement, just simply click to write. If you want to execute multiple statements, use the **BEGIN ... END** compound statement construct.

Example:

```
BEGIN
    set new.capacity = new.capacity + 100;
    set new.amount = new.amount + 100;
END
```

## MySQL Table Options

### Engine

Define the engine of the table.

### Character set

Define the type of the character set for table.

### Collation

Choose the collation for the table.

### Auto Increment

Set/Reset the **Auto Increment** value in the edit field. The **Auto Increment Value** indicates the value for next record.

### ☒ Checksum

Check this option if you want MySQL to maintain a live checksum for all rows.

**Note:** Support *MyISAM* only.

### Row Format

Defines how the rows should be stored.

### Avg. Row Length

An approximation of the average row length for your table. You need to set this only for large tables with variable-size rows.

### Max Rows

The maximum number of rows you plan to store in the table. This is not a hard limit, but rather a hint to the storage engine that the table must be able to store at least this many rows.

### Min Rows

The minimum number of rows you plan to store in the table.

### Key Block Size

This option provides a hint to the storage engine about the size in bytes to use for index key blocks. The engine is allowed to change the value if necessary. A value of 0 indicates that the default value should be used.

## Pack Keys

Set this option to 1 if you want to have smaller indexes. This usually makes updates slower and reads faster. Setting the option to 0 disables all packing of keys. Setting it to **DEFAULT** tells the storage engine to pack only long *CHAR*, *VARCHAR*, *BINARY*, or *VARBINARY* columns.

**Note:** Takes effect only with *MyISAM* tables.

## ☒ Delay Key Write

Check this option if you want to delay key updates for the table until the table is closed.

**Note:** Support *MyISAM* only.

## Data Directory

To specify where the *MyISAM* storage engine should put a table's data file.

## Index Directory

To specify where the *MyISAM* storage engine should put a table's index file.

## Partition

Set the Partition Options.

**Note:** Support from MySQL 5.1 or later.

## MRG\_MYISAM table type

### Union

**UNION** is used when you want to access a collection of identical *MyISAM* tables as one. This works only with *MERGE* tables. You must have *SELECT*, *UPDATE*, and *DELETE* privileges for the tables you map to a *MERGE* table.

### Insert Method

If you want to insert data into a *MERGE* table, you must specify with **INSERT\_METHOD** the table into which the row should be inserted. **INSERT\_METHOD** is an option useful for *MERGE* tables only. Use a value of **FIRST** or **LAST** to have inserts go to the first or last table, or a value of **NO** to prevent inserts.

## **FEDERATED** table type

### **Connection**

To create the local table that will be federated to the remote table. You can create the local table and specify the connection string (containing the server name, login, password) to be used to connect to the remote table using the **Connection** edit box.

The CONNECTION string contains the information required to connect to the remote server containing the table that will be used to physically store the data. The connection string specifies the server name, login credentials, port number and database/table information.

The format the connection string is as follows:

```
scheme://user_name[:password]@host_name[:port_num]/db_name/tbl_name
```

Sample of connection strings:

```
CONNECTION='mysql://username:password@hostname:port/database/tablename'
```

```
CONNECTION='mysql://username@hostname/database/tablename'
```

```
CONNECTION='mysql://username:password@hostname/database/tablename'
```

## **ndbcluster** table type

### **Tablespace**

To specify the tablespace for the storage.

**Note:** Support from MySQL 5.1.6 or later.

### **Storage**

To specify type of storage used (disk or memory), and can be one of **DISK**, **MEMORY**, or **DEFAULT**.

**Note:** Support from MySQL 5.1.6 or later.

## Setting MySQL Table Partition Options

### ☒ Partition By

Select the function that is used to determine the partition: **HASH**, **KEY**, **LINEAR HASH**, **LINEAR KEY**, **RANGE** and **LIST**.

### ☒ Partitions

Set the partition number.





### ☒ Subpartition By

Select the function that is used to determine the subpartition: **Hash** and **Key**.

### ☒ Subpartitions

Set the subpartition number.

## Partition Definition

Use  or  **Partition** to add or delete the partition. Use  or  **Subpartition** to add or delete the subpartition.

## Values

For range partitioning, each partition must include a *VALUES LESS THAN* clause; for list partitioning, you must specify a *VALUES IN* clause for each partition. This is used to determine which rows are to be stored in this partition.

## Engine

Select the storage engine for both partition and subpartition.

## Data Directory

The directory where the data for this partition are to be stored.

## Index Directory

The directory where the indexes for this partition are to be stored.

## Max Rows

The maximum number of rows to be stored in the partition.

## Min Rows

The minimum number of rows to be stored in the partition.

**Tablespace**

Designate a tablespace for the partition. Used for Falcon only.

**Node Group**


Set the Node Group.

**Comment**

Enter the comment for the partition.



## MySQL Views


Views (including updatable views) are implemented in MySQL Server 5.0 and available in binary releases from 5.0.1 and up. Views are useful for allowing users to access a set of relations (tables) as if it were a single table, and limiting their access to just that. Views can also be used to restrict access to rows (a subset of a particular table). For access control to columns, you can also use the sophisticated privilege system in MySQL Server.

Just simply click  to open an object pane for **View**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit, open and delete the selected view.

### Create View

To create a new view

- Select anywhere on the object pane.
- Click the  **New View** from the object pane toolbar.  
or
- Right-click and select  **New View** from the popup menu.
- Edit view properties on the appropriate tabs of the View Designer.

**Hint:** To create new view you can also right-click the Views node of the navigation pane and select the  **New View** from the popup menu.

To create a new view with the same properties as one of the existing views has (using drag and drop method)




**Apply to:** current database {same connection}

- Select the view(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen view(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created view(s) will be named as "viewname\_**copy**".




**Apply to:** different database {same connection}  
different database {different connection} (Data Transfer tool will be activated)

- Select the view(s) for copying in the object pane.
- Drag and drop the chosen view(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel

To create a new view with modification as one of the existing views

- Select the view for modifying in the navigation pane/object pane.
- Right-click and select the  **Design View** from the popup menu.  
or
- Click the  **Design View** from the object pane toolbar.
- Modify view properties on the appropriate tabs of the View Designer.
- Click  **Save As**.

To create a new view with loading from a SQL file

- Select anywhere on the object pane.
- Click the  **New View** from the object pane toolbar.  
or
- Right-click and select  **New View** from the popup menu.
- Click  **Load**.

## Create View Shortcut

To create a view shortcut

- Select the view for editing in the navigation pane/object pane.
- Right-click and select **Create Open View Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.



**Note:** This option is used to provide a convenient way for you to open your view for entering data directly (Grid View/Form View) without activating the main Navicat.





## Edit View

To edit the existing view (manage its SQL definition etc)



- Select the view for editing in the navigation pane/object pane.
- Right-click and select the  **Design View** from the popup menu.  
or
- Click the  **Design View** from the object pane toolbar.
- Edit view properties on the appropriate tabs of the View Designer.

To change the name of the view

- Select the view for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.



## Open View

To open a view (manage view data)

- Select the view for opening in the navigation pane/object pane.
- Right-click and select the  **Open View** from the popup menu or simply double-click the view.  
or
- Click the  **Open View** from the object pane toolbar.

## Delete View

To delete a view

- Select the view for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete View** from the popup menu.  
or
- Click the  **Delete View** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve View Information

To achieve a view information

- Select the view in the navigation pane/object pane.
- Right-click the selected view and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## MySQL View Designer

**View Designer** is the basic Navicat tool for working with views. It allows you to create new view and edit the existing view definition (view name and the SELECT statement it implements).

- [Working with View Builder](#)
- [Editing View SQL Definition](#)
- [Setting Advanced View Properties](#)
- View SQL Preview
- [View Preview](#)
- [View Explain](#)

## **Working with MySQL View Builder (Available only in Full Version)**

**View Builder** allows you to build views visually. It allows you to create and edit views without knowledge of SQL. See Query Builder for details.

## Editing MySQL View SQL Definition

The **Definition** tab allows you to edit the view definition as SQL statement (SELECT statement it implements).

Example:

```
SELECT
    clients.RecordID
FROM
    clients
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see [Editor View and More Features](#).

## Setting Advanced MySQL View Properties

### Algorithm

**Algorithm** is optional and it is a MySQL extension to standard SQL. Algorithm takes three values: **Undefined**, **Merge** or **Temptable**. The default algorithm is Undefined if no Algorithm clause is present. The algorithm affects how MySQL processes the view.

#### Undefined

For **Undefined**, MySQL chooses which algorithm to use. It prefers Merge over Temptable if possible, because Merge is usually more efficient and because a view cannot be updatable if a temporary table is used.

#### Merge

For **Merge**, the text of a statement that refers to the view and the view definition are merged such that parts of the view definition replace corresponding parts of the statement.

#### Temptable

For **Temptable**, the results from the view are retrieved into a temporary table, which then is used to execute the statement.

### Definer

The default **Definer** value is the user who executes the *CREATE VIEW* statement. (This is the same as `DEFINER = CURRENT_USER`.) If a user value is given, it should be a MySQL account in 'user\_name'@'host\_name' format (the same format used in the *GRANT* statement). The user\_name and host\_name values both are required.


### Security

The **SQL SECURITY** characteristic determines which MySQL account to use when checking access privileges for the view when the view is executed. The legal characteristic values are **Definer** and **Invoker**. These indicate that the view must be executable by the user who defined it or invoked it, respectively. The default Security value is Definer.

## Check option

The **Check option** can be given for an updatable view to prevent inserts or updates to rows except those for which the *WHERE* clause in the *select\_statement* is true. The **Local** and **Cascaded** keywords determine the scope of check testing when the view is defined in terms of another view. **Local** restricts the Check option only to the view being defined. **Cascaded** causes the checks for underlying views to be evaluated as well. When neither keyword is given, the default is **Cascaded**.


## MySQL View Preview

To preview the result of the view, click  **Preview** on the toolbar. If the query statement is correct, the **Result** and **Message** tabs will be opened.

The **Result** tab displays the data of the view as a grid and the **Message** tab displays the message log.



## MySQL View Explain

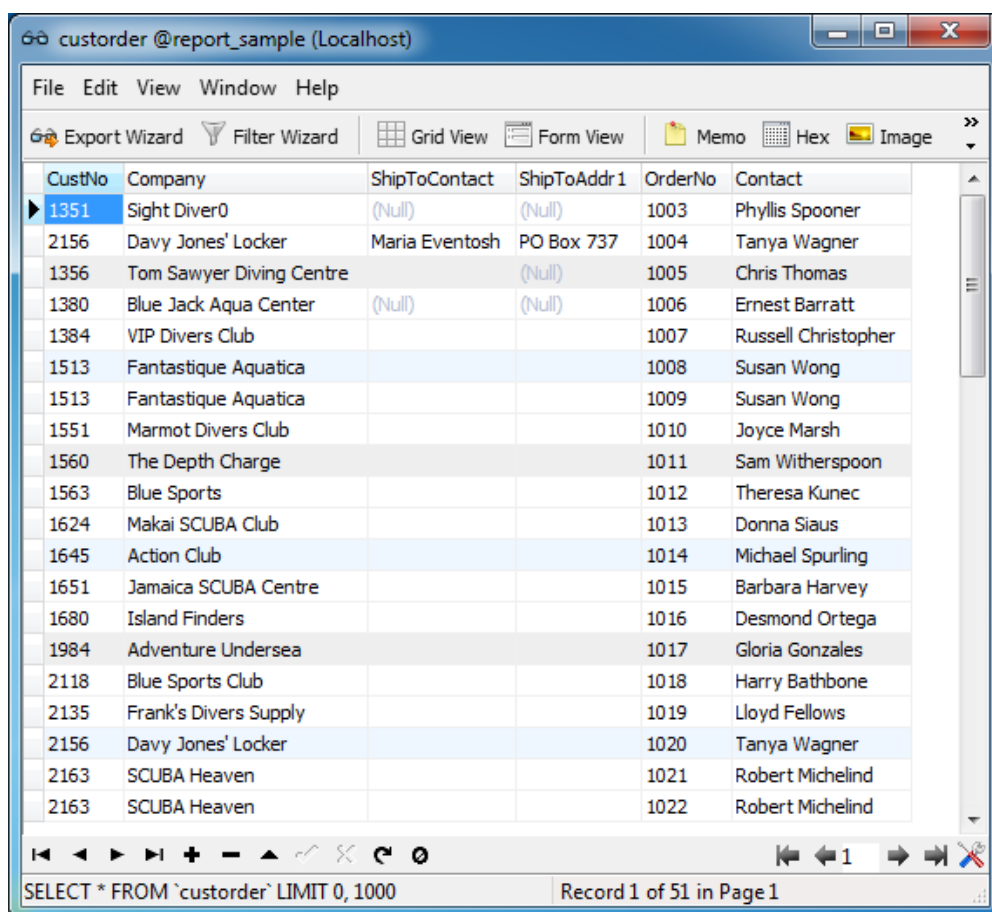
To show the Query Plan of the view, click  **Explain** on the toolbar. If the query statement is correct, the **Explain** tab will show the query plan.

## MySQL View Viewer

**View Viewer** displays the view data as a grid. Data can be displayed in three modes:  **Grid View**,  **Form View** and **Text/Blob View**. See Data View for details.


The toolbars of View Viewer provides the following functions for managing data:

- **Export Data**  
Export data to MS Word, MS Excel, MS Access, TXT, DBF, HTML, SQL, RTF and more.
- **Filter Data**  
Allow you to filter records by creating and applying filter criteria for the data grid.
- **Edit TEXT/BLOB**  
Allow you to view and edit the content of TEXT and BLOB fields.





## MySQL Functions/Procedures


Stored routines (procedures and functions) are supported in MySQL 5.0. A stored routine is a set of SQL statements that can be stored in the server. Once this has been done, clients do not need to keep reissuing the individual statements but can refer to the stored routine instead.

Just simply click  to open an object pane for **Function**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected function/procedure.

### Create Function/Procedure

To create a new function/procedure

- Select anywhere on the object pane.
- Click the  **New Function** from the object pane toolbar.  
or
- Right-click and select  **New Function** from the popup menu.
- Edit function/procedure properties on the appropriate tabs of the Function/Procedure Designer.

**Hint:** To create new function/procedure you can also right-click the Function node of the navigation pane and select the  **New Function** from the popup menu.

To create a new function/procedure with the same properties as one of the existing function/procedure has (using drag and drop method)




**Apply to:** current database {same connection}

- Select the function/procedure(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen function/procedure(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created function/procedure(s) will be named as "function/procedurename\_**copy**".

**Apply to:** different database {same connection}  
different database {different connection} (Data Transfer tool will be activated)



- Select the function/procedure(s) for copying in the object pane.
- Drag and drop the chosen function/procedure(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel

To create a new function/procedure with modification as one of the existing function/procedure

- Select the function/procedure for modifying in the navigation pane/object pane.
- Right-click and select the  **Design Function** from the popup menu or simply double-click the function/procedure.  
or
- Click the  **Design Function** from the object pane toolbar.
- Modify function/procedure properties on the appropriate tabs of the Function/Procedure Designer.
- Click  **Save As**.

## Edit Function/Procedure

To edit the existing function/procedure (manage its definition etc)



- Select the function/procedure for editing in the navigation pane/object pane.
- Right-click and select the  **Design Function** from the popup menu or simply double-click the function/procedure.  
or
- Click the  **Design Function** from the object pane toolbar.
- Edit function/procedure properties on the appropriate tabs of the Function/Procedure Designer.

To change the name of the function/procedure


- Select the function/procedure for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.

## Run Function/Procedure

To run a function/procedure in the navigation pane/object pane



- Select the function/procedure for executing in the navigation pane/object pane.
- Click the  **Execute Function** from the object pane toolbar.  
or
- Right-click and select  **Execute Function** from the popup menu.
- View/edit the returned data on the Result tab.

To run a function/procedure in the Function/Procedure Designer

- Create a new function/procedure/open the existing function/procedure.
- Click  **Run**.
- View/edit the returned data on the Result tab.

## Delete Function/Procedure

To delete a function/procedure


- Select the function/procedure for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Function** from the popup menu.  
or
- Click the  **Delete Function** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Function/Procedure Information

To achieve a function/procedure information

- Select the function/procedure in the navigation pane/object pane.
- Right-click the selected function/procedure and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## MySQL Function Wizard

Click the  **New Function** from the object pane toolbar. The **Function Wizard** will pop up and it allows you to create a procedure/function easily.

- [Setting Routine Type](#)
- [Setting Parameters for Procedure/Function](#)
- [Setting Return Type for Function](#)

You are allowed not to show the **Function Wizard** when create new procedure/function.

**Hint:** Once uncheck the **Show wizard next time**, you can go to Options to enable it.

## Setting MySQL Routine Type

Select the type of the routine: **Procedure** or **Function**

## Setting Parameters for MySQL Procedure/Function

### Procedure

Define the parameter(s) of the procedure. Set the parameter **Mode**, **Name** and **Type** under corresponding columns.

### Function

Define the parameter(s) of the function. Set the parameter **Name** and **Type** under corresponding columns.



## Setting Return Type for MySQL Function

Select the **Return Type** from the list and enter the corresponding information: **Length**, **Decimals**, **Character Set** and/or **Enum**.

**Note:** Only function supports return type.

## MySQL Function/Procedure Designer

**Function/Procedure Designer** is the basic Navicat tool for working with functions/procedures. It allows you to create new function/procedure and edit the existing function/procedure definition.

- [Editing Function/Procedure Definition](#)
- [Setting Advanced Function/Procedure Properties](#)
- Editing Function/Procedure Comment
- Function/Procedure SQL Preview
- [Viewing Function/Procedure Result](#)

## Editing MySQL Function/Procedure Definition

Edit the function/procedure definition under the **Definition** tab. Definition consists of a valid SQL procedure statement. This can be a simple statement such as *SELECT* or *INSERT*, or it can be a compound statement written using *BEGIN* and *END*. Compound statements can contain declarations, loops, and other control structure statements.

Example:

```
BEGIN
    RETURN CONCAT('Hello', name1, ' and ', name2, '!');
END
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see [Editor View and More Features](#).

### Parameter

Define function/procedure parameter.

### Return Type

This text box will be enabled only for creating a function. It indicates the return type of the function.

### Type

Select the stored routines you wish to create from the drop-down list, i.e. **PROCEDURE** and **FUNCTION**.

## Setting Advanced MySQL Function/Procedure Properties

### Security

The **SQL SECURITY** characteristic can be used to specify whether the routine should be executed using the permissions of the user who creates the routine or the user who invokes it. The default value is **Definer**.

### Definer

The default **Definer** value is the user who executes the *CREATE PROCEDURE* or *CREATE FUNCTION* statement. (This is the same as `DEFINER = CURRENT_USER`.) If a user value is given, it should be a MySQL account in 'user\_name'@'host\_name' format (the same format used in the *GRANT* statement). The user\_name and host\_name values both are required.

### Data Access

Several characteristics provide information about the nature of data use by the routine.

#### Contains SQL

Indicates that the routine does not contain statements that read or write data. It is the default if none of these characteristics is given explicitly.

#### No SQL

Indicates that the routine contains no SQL statements.

#### Reads SQL Data

Indicates that the routine contains statements that read data, but not statements that write data.


#### Modifies SQL Data

Indicates that the routine contains statements that may write data.

#### ☒ Deterministic

A procedure or function is considered **deterministic** if it always produces the same result for the same input parameters, and not deterministic otherwise. The default is not deterministic.

## Viewing MySQL Function/Procedure Result

To run the procedure/function click  **Run** on the toolbar. If the SQL statement is correct, the statement will be executed and, if the statement is supposed to return data, the **Result** tab opens with the data returned by the procedure/function. If an error occurs while executing the procedure/function, execution stops, the appropriate error message is displayed.


If the function/procedure requires input parameter, the **Input Parameters** box will popup.

**Note:** The **Result** tab displays the result data as grid.

**Hint:** Navicat supports to return 10 resultsets.



## MySQL Events


MySQL Event Scheduler was added in MySQL 5.1.6. MySQL Events are tasks that run according to a schedule. Therefore, we sometimes refer to them as scheduled events. When you create an event, you are creating a named database object containing one or more SQL statements to be executed at one or more regular intervals, beginning and ending at a specific date and time. Conceptually, this is similar to the idea of the Windows Task Scheduler.

Just simply click  to open an object pane for **Event**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected event.

### Create Event

To create a new event

- Select anywhere on the object pane.
- Click the  **New Event** from the object pane toolbar.  
or
- Right-click and select  **New Event** from the popup menu.
- Edit event properties on the appropriate tabs of the Event Designer.

**Hint:** To create new event you can also right-click the Event node of the navigation pane and select the  **New Event** from the popup menu.

To create a new event with the same properties as one of the existing event has (using drag and drop method)




**Apply to:** current database {same connection}

- Select the event(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen event(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created event(s) will be named as "eventname\_**copy**".

**Apply to:** different database {same connection}  
different database {different connection} (Data Transfer tool will be activated)



- Select the event(s) for copying in the object pane.
- Drag and drop the chosen event(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel

To create a new event with modification as one of the existing event

- Select the event for modifying in the navigation pane/object pane.
- Right-click and select the  **Design Event** from the popup menu or simply double-click the event.  
or
- Click the  **Design Event** from the object pane toolbar.
- Modify event properties on the appropriate tabs of the Event Designer.
- Click  **Save As**.

## Edit Event

To edit the existing event (manage its definition etc)



- Select the event for editing in the navigation pane/object pane.
- Right-click and select the  **Design Event** from the popup menu or simply double-click the event.  
or
- Click the  **Design Event** from the object pane toolbar.
- Edit event properties on the appropriate tabs of the Event Designer.

To change the name of the event

- Select the event for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.

## Delete Event

To delete an event

- Select the event for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Event** from the popup menu.  
or
- Click the  **Delete Event** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Event Information

To achieve an event information

- Select the event in the navigation pane/object pane.
- Right-click the selected event and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.



## MySQL Event Designer

**Event Designer** is the basic Navicat tool for working with events. It allows you to create new event and edit the existing event definition.

- [Editing Event Definition](#)
- [Setting Advanced Event Properties](#)
- Editing Event Comment
- Event SQL Preview

## Editing MySQL Event Definition

Edit the event definition under the **Definition** tab. Definition consists of a valid SQL statement. This can be a simple statement such as *SELECT* or *INSERT*, or it can be a compound statement written using *BEGIN* and *END*. Compound statements can contain declarations, loops, and other control structure statements.

**Hint:** To customize the view of the editor and find out more features for sql editing, see [Editor View and More Features](#).

### Definer

Specifies the user account to be used when checking access privileges at event execution time. The default DEFINER value is the user who executes the *CREATE EVENT* statement. (This is the same as DEFINER = CURRENT\_USER.) If a user value is given, it should be a MySQL account in 'user\_name'@'host\_name' format (the same format used in the GRANT statement). The user\_name and host\_name values both are required.

### STATUS

You can create an event but keep it from being active using the *DISABLE* keyword. Alternatively, you may use *ENABLE* to make explicit the default status, which is active.

### ON COMPLETION

Normally, once an event has expired, it is immediately dropped. You can override this behavior by specifying *ON COMPLETION PRESERVE*. Using *ON COMPLETION NOT PRESERVE* merely makes the default non-persistent behavior explicit.

## Setting Advanced MySQL Event Properties

Edit the *ON SCHEDULE* clause under **Schedule** tab. The *ON SCHEDULE* clause determines when, how often, and for how long the SQL statement defined for the event repeats. This clause takes one of two forms:

### **AT**

*AT timestamp* is used for a one-time event. It specifies that the event executes one time only at the date and time, given as the *timestamp*, which must include both the date and time, or must be an expression that resolves to a datetime value.

Use **+INTERVAL** to create an event which occurs at some point in the future relative to the current date and time.

### **EVERY**

For actions which are to be repeated at a regular interval, you can use an *EVERY* clause which followed by an *interval*. (**+INTERVAL** is not used with *EVERY*.)

#### **STARTS**

An *EVERY* clause may also contain an optional *STARTS* clause. *STARTS* is followed by a *timestamp* value which indicates when the action should begin repeating, and may also use **+INTERVAL** interval in order to specify an amount of time "from now".

For example: ***EVERY 3 MONTH STARTS CURRENT\_TIMESTAMP + 1 WEEK***  
means "every three months, beginning one week from now".

#### **ENDS**

An *EVERY* clause may also contain an optional *ENDS* clause. The *ENDS* keyword is followed by a *timestamp* value which tells MySQL when the event should stop repeating. You may also use **+INTERVAL** interval with *ENDS*.

For Instance: **EVERY 12 HOUR STARTS CURRENT\_TIMESTAMP + INTERVAL 30 MINUTE ENDS CURRENT\_TIMESTAMP + INTERVAL 4 WEEK** is equivalent to "every twelve hours, beginning thirty minutes from now, and ending four weeks from now".

P.S.	The <i>timestamp</i> must be in the future - you cannot schedule an event to take place in the past.
	<p>The <i>interval</i> portion consists of two parts, a quantity and a *unit of time.</p> <p>*YEAR   QUARTER   MONTH   DAY   HOUR   MINUTE    WEEK   SECOND   YEAR_MONTH   DAY_HOUR    DAY_MINUTE    DAY_SECOND   HOUR_MINUTE   HOUR_SECOND    MINUTE_SECOND</p>

## Oracle Database Object Management

The following list contains the most common Oracle database objects supported by Navicat.

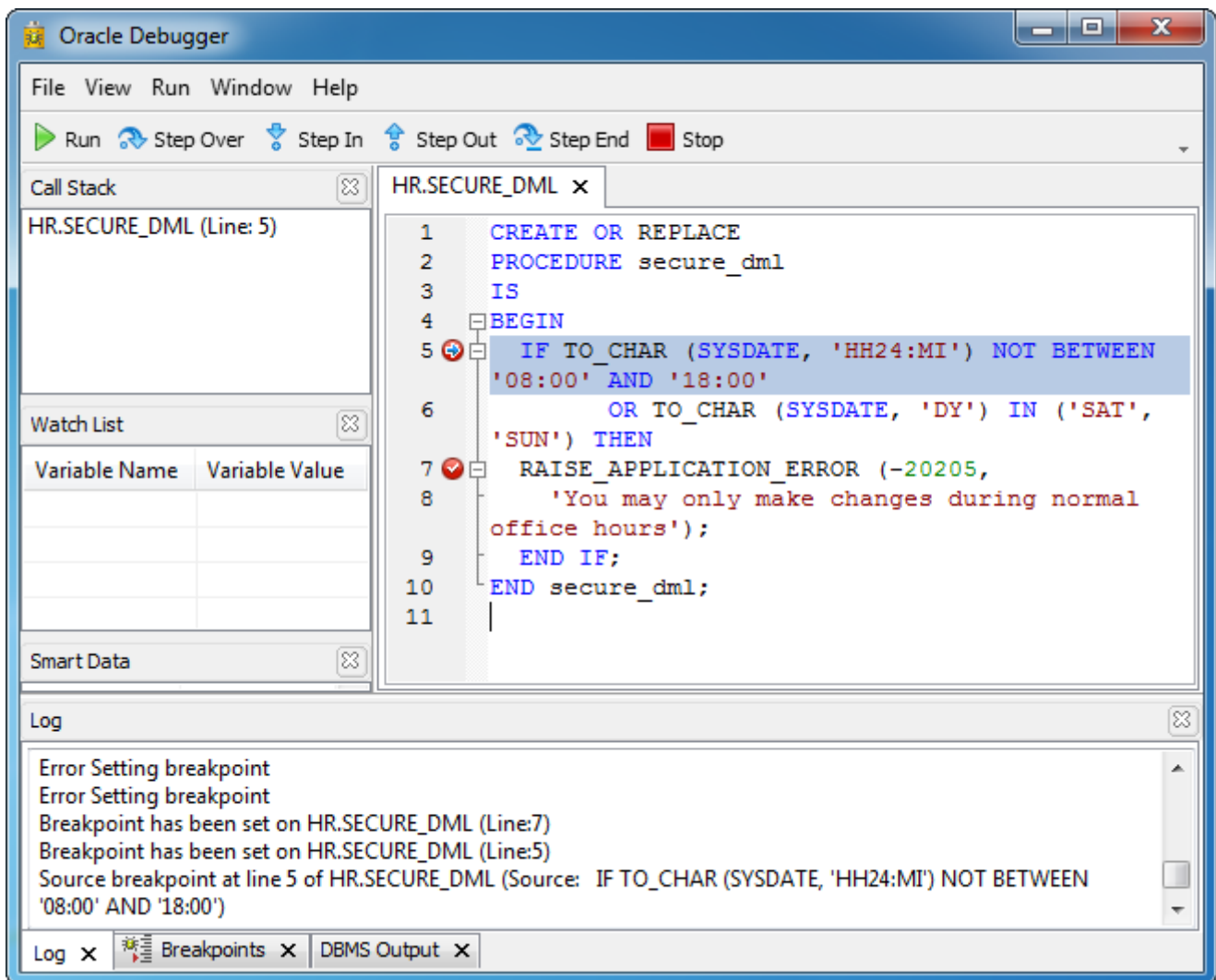
**Hint:** Oracle interprets non-quoted object identifiers as uppercase. In Navicat, all objects identifier will be quoted. That is, Navicat saves exactly what you have inputted.

- [Tables](#)
- [Views](#)
- [Functions/Procedures](#)
- [Database Links](#)
- [Indexes](#)
- [Java](#)
- [Materialized Views](#)
- [Materialized View Logs](#)
- [Packages](#)
- [Sequences](#)
- [Synonyms](#)
- [Triggers](#)
- [Types](#)
- [XML Schemas](#)
- [Recycle Bin](#)
- [Directories](#)
- [Tablespaces](#)
- [Public Database Links](#)
- [Public Synonyms](#)


## Oracle Debugger (Available only in Full Version)

Navicat provides Oracle PL/SQL debugger for debugging Oracle functions, procedures, packages and queries.

- [Code Window](#)
- [Views](#)
- [Oracle Debugger Toolbar](#)




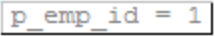
## Oracle Debugger Code Window

The **Code Window** shows the code of the procedure/function. You can add/remove breakpoints for debugging by clicking  in the grey area beside each statement.

To add a variable to the watch list, simply right-click the highlighted code and choose **Add to watch list**.

To show the debug tips, simply mouse-over the code.

```

1  CREATE OR REPLACE
2  PROCEDURE add_job_history
3  ( p_emp_id          job_history.employee_id%type
4    , p_start_date     job_history.start_date%type
5    , p_end_date       job_history.end_date%type
6    , p_job_id         job_history.job_id%type
7    , p_department_id  job_history.department_id%type
8  )
9  IS
10 BEGIN
11  INSERT INTO job_history (employee_id, start_date,
    end_date,
12                                job_id, department_id)
13     VALUES (p_emp_id, p_start_date, p_end_date, p_job_id,
    p_department_id);  p_emp_id = 1
14 END add_job_history;
  
```

## Oracle Debugger Views

Under menu **View**, you can choose to show/hide the following view windows.

You can choose View -> **Default Layout** to restore the Layout to the default one.

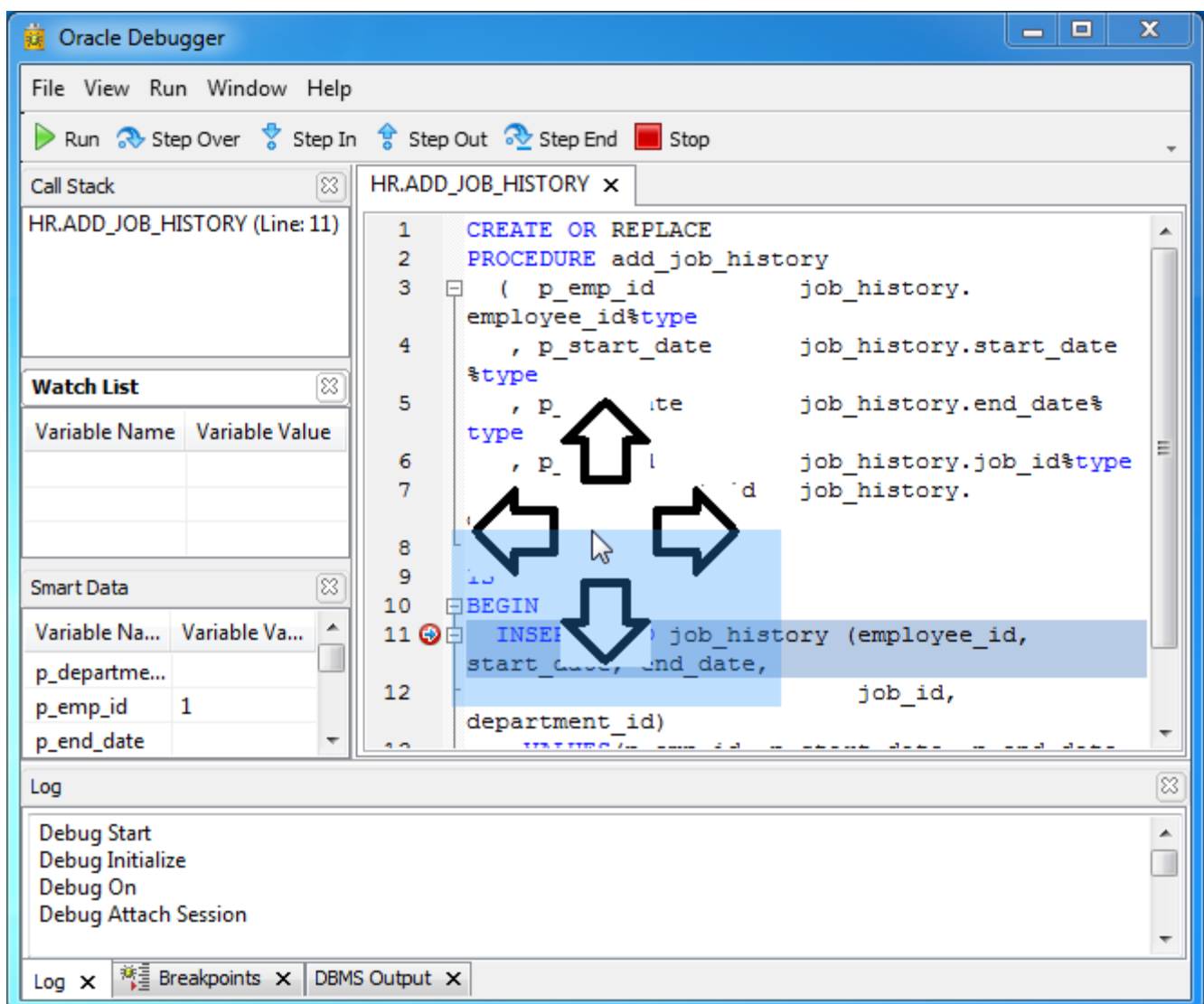
- [Breakpoints](#)
- [Call Stack](#)
- [DBMS Output](#)
- [Log](#)
- [Smart Data](#)
- [Watch List](#)



## Repositioning Oracle Debugger Views

To re-arrange the workspace items, click on any pane in the workspace, then hold down your cursor and drag the pane to the desired area, and release the cursor. The selected pane will appear in its new position.

**Hint:** As you drag a pane to its new position, a **Pane Sticker** will appear and a bright blue indicator will mark the insertion point.



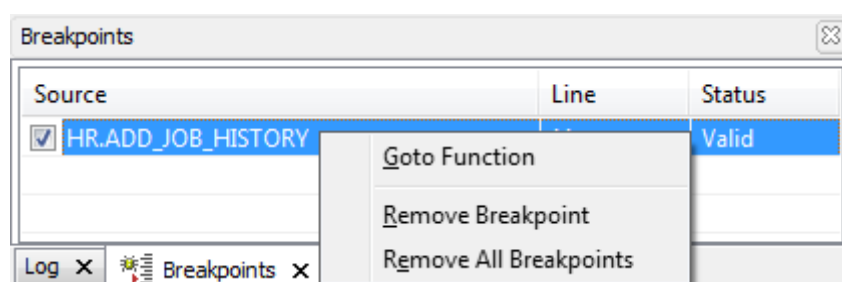
## Oracle Debugger Breakpoints

The **Breakpoints** view displays all the breakpoints which allowing you to delete, enable or disable breakpoints.

To enable/disable a breakpoint, simply check/uncheck the check box.

Also, you can delete a breakpoint or all breakpoints, simply right-click a breakpoint and choose **Remove Breakpoint** or **Remove All Breakpoints**.

To jump to the line of a breakpoint, right-click it and choose **Goto Function**.



## Oracle Debugger Call Stack

The **Call Stack** view displays the procedure or function calls of the current line.

To jump to a procedure or function, right-click it and choose **Goto Function**.

## Oracle Debugger DBMS Output

The **DBMS Output** view displays the results after the function or procedure has completed the execution.

## Oracle Debugger Log

The **Log** view shows the message log when debugging the code.

## Oracle Debugger Smart Data

The **Smart Data** view displays information about the variables associated with breakpoints.

To add a variable to the watch list, simply right-click a variable and choose **Add to watch list**.

You can adjust the value of a watch variable by simply right-click the variable and choose **Adjust Value**.

## Oracle Debugger Watch List

The **Watch List** view displays information about the variables being watched, allowing you to add, delete or edit watch variables.

To add a watch variable, simply right-click anywhere of Watch List view and choose **Add Variable**. Then, enter the **Variable Name**. Also, you can right-click the highlighted code in the Code Window or the variable in the Smart Data view and choose **Add to watch list**.


You can adjust the value of a watch variable by right-click the variable and choose **Adjust Value**.

To delete a watch variable or all watch variables, simply right-click a variable and choose **Remove Variable** or **Remove All Variable**.


## Oracle Debugger Toolbar

You can perform the most commonly used actions for debugging on the toolbar or menu:


### **Run**

Start running code in debug mode by clicking  **Run** or pressing **F9**. The debugger executes your code until the end of the code or the next breakpoint is reached.


### **Step Over**

While execution of your code is paused, you can resume it by clicking  **Step Over** or pressing **F8**. Then, the current line will be executed. If the line is a procedure or function call, it will bypass the procedure or function. The counter will then move to the next line of code.


### **Step In**

While execution of your code is paused, you can resume it by clicking  **Step In** or pressing **F7**. Then, it executes the current line. If the line is a procedure or function call, the counter goes to the first statement in the procedure or function. Otherwise, the counter will move to the next line of code.


### **Step Out**

While execution of your code is paused, you can resume it by clicking  **Step Out** or pressing **Shift+F7**. Then, the remaining part of the code within the current procedure or function will be executed. Subsequently, the counter will jump to the line which is just after the caller of the method.

### **Step End**

While execution of your code is paused, you can resume it by clicking  **Step End**. Then, the counter will jump to the last line of the procedure or function.

### **Stop**

While execution of your code is paused, you can stop stepping the code by clicking  **Stop**. Then, the execution will stop and cannot resume it.



## Editing Oracle Physical Attributes/Default Storage Characteristics

### **% Free**

Specify a whole number representing the percentage of space in each data block of the database object reserved for future updates to rows of the object. The value must be from 0 to 99. A value of 0 means that the entire block can be filled by inserts of new rows. The default value is 10.

### **% Used**

Specify a whole number representing the minimum percentage of used space that Oracle maintains for each data block of the database object. A block becomes a candidate for row insertion when its used space falls below this value. The value must be from 0 to 99 and defaults to 40.

### **Initial Transactions**

Specify the initial number of concurrent transaction entries allocated within each data block allocated to the database object. This value can range from 1 to 255 and defaults to 1.

**Note:** The default value for an index is 2.

### **Maximum Transactions**

Specify the maximum number of concurrent update transactions allowed for each data block in the segment.

## **Storage**

### **Initial**

Specify the size of the first extent of the object. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.

### **Next**

Specify the size of the next extent to be allocated to the object. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.

### **Min Extents**

Specify the total number of extents to allocate when the object is created.

## Max Extents

Specify the total number of extents, including the first, that Oracle can allocate for the object. Check **Unlimited** if you want extents to be allocated automatically as needed.

## Max Size

Specify the maximum size of the storage element. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes. Check **Unlimited** if you do not want to limit the disk space of the storage element.

## % Increase

Specify the percent by which the third and subsequent extents grow over the preceding extent. The default value is 50.

## Freelists

For objects other than tablespaces and rollback segments, specify the number of free lists for each of the free list groups for the table, partition, cluster, or index.

## Freelist Group

Specify the number of groups of free lists for the database object you are creating. The default and minimum value for this parameter is 1.

## Optimal

Specify an optimal size for a rollback segment. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes. Check **Null** for no optimal size for the rollback segment.

## Buffer Pool

### DEFAULT

Choose this to indicate the default buffer pool. This is the default for objects not assigned to KEEP or RECYCLE.

### KEEP


Choose this to put blocks from the segment into the KEEP buffer pool. Maintaining an appropriately sized KEEP buffer pool lets Oracle retain the schema object in memory to avoid I/O operations. KEEP takes precedence over any NOCACHE clause you specify for a table, cluster, materialized view, or materialized view log.

## **RECYCLE**

Choose this to put blocks from the segment into the RECYCLE pool. An appropriately sized RECYCLE pool reduces the number of objects whose default pool is the RECYCLE pool from taking up unnecessary cache space.




## Oracle Tables


Relational databases use tables to store data. All operations on data are done on the tables themselves or produce another table as the result. A table is a set of rows and columns, and their intersections are fields. From a general perspective, columns within a table describe the name and type of data that will be found by row for that column's fields. Rows within a table represent records composed of fields that are described from left to right by their corresponding column's name and type. Each field in a row is implicitly correlated with each other field in that row.

Just simply click  to open an object pane for **Table**. A right-click displays the popup menu or by using the object pane toolbar, allowing you to create new, edit, open and delete the selected table.

### Create Table

To create a new table

- Select anywhere on the object pane.
- Click the  **New Table** from the object pane toolbar together with the  down arrow to choose the type **Normal** / **External** / **Index Organized**.  
or
- Right-click and select  **New Table** to choose the type **Normal** / **External** / **Index Organized** from the popup menu.
- Edit table properties and fields on the appropriate tabs of the Table Designer.

**Hint:** To create new table you can also right-click the Tables node of the navigation pane and select the  **New Table** from the popup menu.

To create a new table with the same properties as one of the existing tables has (using popup menu)

**Apply to:** current schema {same connection}

- Select the table(s) for copying in the navigation pane/object pane.
- Right-click and select the **Duplicate Table** from the popup menu.
- The newly created table(s) will be named as "tablename\_**copy**".



To create a new table with the same properties as one of the existing tables has (using drag and drop method)

**Apply to:** current schema {same connection}




- Select the table(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen table(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created table(s) will be named as "tablename\_**copy**"

**Apply to:** different schema {same connection}

different schema {different connection (same or cross server type)} (Data Transfer tool will be activated)

- Select the table(s) for copying in the object pane.
- Drag and drop the chosen table(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel

To create a new table with modification as one of the existing tables

- Select the table for modifying in the navigation pane/object pane.
- Right-click and select the  **Design Table** from the popup menu.  
or
- Click the  **Design Table** from the object pane toolbar.
- Modify table properties and fields on the appropriate tabs of the Table Designer.
- Click  **Save As**.

## Create Table Shortcut



To create a table shortcut

- Select the table for editing in the navigation pane/object pane.
- Right-click and select **Create Open Table Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.

**Note:** This option is used to provide a convenient way for you to open your table for entering data directly (Grid View/Form View) without activating the main Navicat.

## Edit Table

To edit the existing table (manage its fields, indexes, foreign keys and triggers etc)



- Select the table for editing in the navigation pane/object pane.
- Right-click and select the  **Design Table** from the popup menu.  
or
- Click the  **Design Table** from the object pane toolbar.
- Edit table properties and fields on the appropriate tabs of the Table Designer.


To change the name of the table

- Select the table for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.


## Open Table (manage table data)

To open a table

- Select the table for opening in the navigation pane/object pane.
- Right-click and select the  **Open Table** from the popup menu or simply double-click the table.  
or
- Click the  **Open Table** from the object pane toolbar.

**Note:** This option is only applied if you do wish Navicat loads all your images while opening the table. To open the graphical table with faster performance, use  **Open Table (Quick)** below.

To open a table with graphical fields

- Select the table for opening in the navigation pane/object pane.
- Right-click and select the  **Open Table (Quick)** from the popup menu.

**Note:** Faster performance for opening the graphical table, as BLOB fields (images) will not be loaded until you click on the cell.

## Empty Table

To empty a table

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Empty Table** from the popup menu.

**Note:** This option is only applied when you wish to clear all the existing records without resetting the auto-increment value. To reset the auto-increment value while emptying your table, use **Truncate Table** below.



## Truncate Table

To truncate a table

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Truncate Table** from the popup menu.

## Delete Table

To delete a table

- Select the table for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Table** from the popup menu.  
or
- Click the  **Delete Table** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Table Information

To achieve a table information

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.



## Oracle Normal Tables

Tables are the basic unit of data storage in an Oracle database. Data is stored in rows and columns. You define a table with a table name and set of columns.

In a normal (heap-organized) table, data is stored as an unordered collection (heap).

## Table Designer for Oracle Normal Tables (/Index Organized Tables)

**Table Designer** is the basic Navicat tool for working with tables. It allows you to create, edit and drop table's fields, indexes, foreign keys, and much more.

Note that **Table Designer** for **Index Organized Tables** differs from **Normal Tables** only on the [Options](#) tab.



- [Managing Table Fields](#)
- [Managing Table Indexes](#)
- [Managing Table Foreign Keys](#)
- [Managing Table Uniques](#)
- [Managing Table Checks](#)
- [Managing Table Triggers](#)
- [Managing Table Options](#)
- Managing Table Comment
- Table SQL Preview

## Oracle Table Fields

Table fields are managed on the **Fields** tab of the Table Designer. Just simply click a field for editing. A right-click displays the popup menu or by using the field toolbar, allowing you to create new and drop the selected field.

### Add Field

To add a field to the table

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click and select the  **Add Field** from the popup menu or click the  **Add Field** from the toolbar.
- Edit field properties.

To add a new field with modification as one of the existing fields

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Select field.
- Right-click and select the **Duplicate Field** from the popup menu.
- Edit field properties.



### Edit Field

To edit the table field


- Open the table in the Table Designer.
- Open the **Fields** tab.
- Simply click on the field to edit.

## Delete Field

To delete the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click and select the  **Delete Field** from the popup menu or click the  **Delete Field** from the toolbar.
- Confirm deleting in the dialog window.

## Setting Oracle Table Field Properties

Name	Type	Length	Scale	Allow Null	
▶ EMPLOYEE_ID	NUMBER	6	0	<input type="checkbox"/>	 1
FIRST_NAME	VARCHAR2	20	0	<input checked="" type="checkbox"/>	
LAST_NAME	VARCHAR2	25	0	<input type="checkbox"/>	
EMAIL	VARCHAR2	25	0	<input type="checkbox"/>	

### Name

The Name is a descriptive identifier for a field that can be up to 30 bytes by default (letters or numbers) including spaces. The names should be descriptive enough that anyone can easily identify them when viewing or editing records. For example, LastName, FirstName, StreetAddress, or HomePhone.

Use the **Name** edit box to set the field name. Note that the name of the field must be unique among all the field names in the table.

### Type

After you name a field, you choose a data type for the data to be contained in the field. When you choose a field's data type, you are deciding:

- What kind of values to allow in the field. You cannot store text in field with the **Numeric** data type.
- How much storage space Oracle is to set aside for the data in that field.
- What types of operations can be performed on the values in that field.

The **Type** dropdown list defines the type of the field data.

The following table shows all the built-in general-purpose data types.

Name	Description
CHAR	fixed-length character strings
NCHAR	fixed-length Unicode character data
VARCHAR2	variable-length character strings
VARCHAR	variable-length character strings
NVARCHAR2	variable-length Unicode character data
CLOB	database character set data
NCLOB	Unicode national character set data

LONG	variable-length character data containing up to 2 gigabytes of information
NUMBER	fixed and floating-point numbers
DATE	point-in-time values (dates and times)
INTERVAL DAY TO SECOND	period of time in terms of days, hours, minutes, and seconds
INTERVAL YEAR TO MONTH	stores a period of time using the YEAR and MONTH datetime fields
TIMESTAMP	point-in-time values (dates and times) (includes fractional seconds)
TIMESTAMP WITH TIME ZONE	TIMESTAMP with explicit time zone information
BLOB	unstructured binary data in the database
BFILE	unstructured binary data in operating-system files outside the database
RAW	can be indexed and is used for data that is not to be interpreted by Oracle Database
LONG RAW	cannot be indexed and is used for data that is not to be interpreted by Oracle Database
ROWID	the address (rowid) of every row in the database
CHARACTER	=CHAR <sup>1</sup>
CHARACTER VARYING	=VARCHAR2 <sup>1</sup>
CHAR VARYING	=VARCHAR2 <sup>1</sup>
NATIONAL CHARACTER	=NCHAR <sup>1</sup>
NATIONAL CHAR	=NCHAR <sup>1</sup>
NATIONAL CHARACTER VARYING	=NVARCHAR2 <sup>1</sup>
NATIONAL CHAR VARYING	=NVARCHAR2 <sup>1</sup>
NCHAR VARYING	=NVARCHAR2 <sup>1</sup>
NUMERIC	=NUMBER <sup>1</sup>
DECIMAL	=NUMBER <sup>1</sup>
INTEGER	=NUMBER(38) <sup>1</sup>
INT	=NUMBER(38) <sup>1</sup>
SMALLINT	=NUMBER(38) <sup>1</sup>
FLOAT	=FLOAT(126) <sup>1</sup>
DOUBLE PRECISION	=FLOAT(126) <sup>1</sup>
REAL	=FLOAT(63) <sup>1</sup>

**Note:** These are ANSI datatypes and datatypes from the IBM products SQL/DS and DB2. Oracle recognizes these datatypes and converts them to the equivalent Oracle datatype. [Click here](#) for detailed description on these datatypes.

## Length and Scale

Use the **Length** edit box to define the **precision** (total number of digits) of the field and use **Scale** edit box to define the **scale** (number of digits to the right of the decimal point) for **numeric** column.

**Note:** Be careful when shortening the field length as it might result in data loss.

How scale factors affect numeric data storage:

Input Data	Specified As	Stored As
7,456,123.89	NUMBER	7456123.89
7,456,123.89	NUMBER(*,1)	7456123.9
7,456,123.89	NUMBER(9)	7456124
7,456,123.89	NUMBER(9,2)	7456123.89
7,456,123.89	NUMBER(9,1)	7456123.9
7,456,123.89	NUMBER(6)	not accepted, exceeds precision
7,456,123.89	NUMBER(7,-2)	7456100

## Allow Null

Allow the NULL values for the field.

## Primary Key

A **Primary Key** is a single field or combination of fields that uniquely defines a record. None of the fields that are part of the primary key can contain a null value.

## Primary Key Name

Right-click and select **Primary Key Name** from the popup menu to enter the primary key constraint name.

## Setting Other Oracle Table Field Properties

For **INTERVAL DAY TO SECOND** data type:

### Leading Field Precision

Set the number of digits in the leading field. Accepted values are 0 to 9. The default is 2.

### Fractional Seconds Precision

Set the number of digits in the fractional part of the SECOND datetime field. Accepted values are 1 to 9.

For **INTERVAL YEAR TO MONTH** data type:

### Year Precision

Set the number of digits in the year. The default is 2.

For **TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE** data types:

### Fractional Seconds Precision

Set the number of digits in the fractional part of the SECOND datetime field. Accepted values are 1 to 9.

For **CHAR, VARCHAR2** data types:

### Unit

Set the unit either in BYTE or CHAR.

For **COMPLEX** data types:

### Object Schema

Set the object schema for the field.

### Object Type

Set the object type for the field.

For most data types:

### Default

Set the default value for the field.



For all data types:

**Comment**

Set any optional text describing the current field.

## Oracle Table Indexes



Indexes are optional structures associated with tables and clusters. You can create indexes on one or more columns of a table to speed SQL statement execution on that table. An Oracle Database index provides a faster access path to table data. Indexes are the primary means of reducing disk I/O when properly used.

You can create many indexes for a table as long as the combination of columns differs for each index. You can create more than one index using the same columns if you specify distinctly different combinations of the columns.

Table indexes are managed on the **Indexes** tab of the Table Designer. Just simply click/double-click an index field for editing. A right-click displays the popup menu or using the index toolbar, allowing you to create new, edit and delete the selected index field.

### Add Index

To add a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Right-click and select the  **Add Index** from the popup menu or click the  **Add Index** from the toolbar.
- Edit index properties.



### Edit Index

To edit a table index


- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Just simply click/double-click on the index to edit.

## Delete Index


To delete a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Right-click on the index to delete and select the  **Delete Index** from the popup menu or click the  **Delete Index** from the toolbar.
- Confirm deleting in the dialog window.

## Setting Oracle Table Index Properties

Name	Fields	Index Type
EMP_DEPARTMENT_IX	DEPARTMENT_ID ASC	Non-Unique
▶ EMP_JOB_IX	JOB_ID ASC	 Non-Unique

Use the **Name** edit box to set the index name.

To include field(s) in the index, just simply double-click the **Fields** field or click  to open the editor for editing.

The **Index Type** dropdown list defines the type of the table index. Oracle Database provides several indexing schemes.

### Non-unique

Non-unique indexes do not impose the restriction of unique indexes on the column values.

### Unique

Unique indexes guarantee that no two rows of a table have duplicate values in the key column (or columns).

### Bitmap

In a bitmap index, a bitmap for each key value is used instead of a list of rowids.

### ☒ Parallel With Degree

Parallel indexing can improve index performance when you have a large amount of data, and have multiple CPUs. Enter the degree that determines the number of separate indexing processes.

### Tablespace

The tablespace in which to create the index. An index can be created in the same or different tablespace as the table it indexes.

## Schema

The schema in which to create the index.

### Note:

To create an index in your own schema, at least one of the following conditions must be true:

- The table or cluster to be indexed is in your own schema.
- You have INDEX privilege on the table to be indexed.
- You have CREATE ANY INDEX system privilege.

To create an index in another schema, all of the following conditions must be true:

- You have CREATE ANY INDEX system privilege.
- The owner of the other schema has a quota for the tablespaces to contain the index or index partitions, or UNLIMITED TABLESPACE system privilege.



## Oracle Table Foreign Keys

A foreign key specifies that the values in a column (or a group of columns) must match the values appearing in some row of another table. We say this maintains the referential integrity between two related tables.

Foreign Keys are managed on the **Foreign Keys** tab of the Table Designer. Just simply click/double-click a foreign key field for editing. A right-click displays the popup menu or using the foreign key toolbar, allowing you to create new, edit and delete the selected foreign key field.

### Add Foreign Key

To add a foreign key

- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Right-click and select the  **Add Foreign Key** from the popup menu or click the  **Add Foreign Key** from the toolbar.
- Edit foreign key properties.



### Edit Foreign Key

To edit a foreign key


- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Just simply click/double-click on the foreign key to edit.

### Delete Foreign Key

To delete a foreign key


- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Right-click on the foreign key to delete and select the  **Delete Foreign Key** from the popup menu or click the  **Delete Foreign Key** from the toolbar.
- Confirm deleting in the dialog window.

## Setting Oracle Table Foreign Key Properties

Name	Reference Schema	Reference Table	Reference Constraint	Reference Fields	Fields	On Delete	Enable
EMP_DEPT_FK	HR	DEPARTMENTS	DEPT_ID_PK	DEPARTMENT_ID	DEPARTMENT_ID		<input checked="" type="checkbox"/>
▶ EMP_JOB_FK	HR	JOBS	JOB_ID_PK	JOB_ID	JOB_ID  		<input checked="" type="checkbox"/>

Use the **Name** edit box to enter a name for the new key and then select a table field to include in the key from the **Fields** group.

Use the **Reference Schema**, **Reference Table** and **Reference Constraint** dropdown lists to select a foreign schema, table and constraint respectively.

To include field(s) to the key, just simply double-click the **Fields** field or click  to open the editor(s) for editing.

The **On Delete** dropdown list defines the type of the actions to be taken.

### No Action (default)

Referenced key values will not be updated or deleted.

### Cascade

Delete any rows referencing the deleted row, or update the value of the referencing column to the new value of the referenced column, respectively.

### Set Null

Set the referencing column(s) to null.

### Enable

You can choose whether to enable / disable the foreign key constraint by checking / unchecking the box.



## Oracle Table Uniques

Unique constraints ensure that the data contained in a column or a group of columns is unique with respect to all the rows in the table.

Uniques are managed on the **Uniques** tab of the Table Designer. Just simply click/double-click an unique field for editing. Using the toolbar below, allowing you to create new, edit and delete the selected unique field.

### Add Unique

To add an unique

- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Right-click and select the  **Add Unique** from the popup menu or click the  **Add Unique** from the toolbar.
- Edit unique properties.



### Edit Unique

To edit an unique

- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Just simply click on the unique to edit.


### Delete Unique

To delete an unique

- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Right-click on the unique to delete and select the  **Delete Unique** from the popup menu or click the  **Delete Unique** from the toolbar.
- Confirm deleting in the dialog window.




## Oracle Table Setting Unique Properties

Name	Fields	Enable
▶ EMP_EMAIL_UK	EMAIL	 <input checked="" type="checkbox"/>

Use the **Name** edit box to set the unique name.

### Fields

To set field(s) as unique, just simply double-click the **Fields** field or click  to open the editor(s) for editing.

### ☒ Enable

You can choose whether to enable / disable the unique constraint by checking / unchecking the box.



## Oracle Table Checks

A check constraint is the most generic constraint type. It allows you to specify that the value in a certain column must satisfy a Boolean (truth-value) expression.

Checks are managed on the **Checks** tab of the Table Designer. Just simply click/double-click a check field for editing. Using the check toolbar, allowing you to create new, edit and delete the selected check field.

### Add Check

To add a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Right-click and select the  **Add Check** from the popup menu or click the  **Add Check** from the toolbar.
- Edit check properties.



### Edit Check

To edit a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Just simply click on the check to edit.

### Delete Check

To delete a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Right-click on the check to delete and select the  **Delete Check** from the popup menu or click the  **Delete Check** from the toolbar.
- Confirm deleting in the dialog window.

## Setting Oracle Table Check Properties

Use the **Name** edit box to set the check name.

### Check

Set the condition for checking, e.g. "field\_name1 > 0 AND field\_name2 > field\_name1" in the **Check** edit box. A check constraint specified as a column constraint should reference that column's value only, while an expression appearing in a table constraint may reference multiple columns.

### Definition

Type in the definition for the check constraint.

### ☒ **Enable**

You can choose whether to enable / disable the check constraint by checking / unchecking the box.



## Oracle Table Triggers

A trigger is a specification that the database should automatically execute a particular function whenever a certain type of operation is performed. Triggers can be defined to execute either before or after any INSERT, UPDATE, or DELETE operation, either once per modified row, or once per SQL statement.

Triggers are managed on the **Triggers** tab of the Table Designer. Just simply click a trigger field for editing. A right-click displays the popup menu or using the trigger toolbar, allowing you to create new, edit and delete the selected trigger field.

### Add Trigger

To add a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Right-click and select the  **Add Trigger** from the popup menu or click the  **Add Trigger** from the toolbar.
- Edit trigger properties.



### Edit Trigger

To edit a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Just simply click on the trigger to edit.

### Delete Trigger

To delete a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Right-click on the trigger to delete and select the  **Delete Trigger** from the popup menu or click the  **Delete Trigger** from the toolbar.
- Confirm deleting in the dialog window.

## Setting Oracle Table Trigger Properties

### Name

Set the trigger name.

### Compound

Check to set the trigger as a compound trigger.

**Note:** Support from Oracle 11g or later.

### Row trigger

Check to set the trigger as a row trigger.

### Fires

Specify the trigger timing whether the trigger action is to be run before or after the triggering statement.

*INSERT* - fires the trigger whenever an INSERT statement adds a row to a table or adds an element to a nested table.

*UPDATE* - fires the trigger whenever an UPDATE statement changes a value in one of the columns specified in **Update of Fields**. If no **Update of Fields** are present, the trigger will be fired whenever an UPDATE statement changes a value in any column of the table or nested table.

*DELETE* - fires the trigger whenever a DELETE statement removes a row from the table or removes an element from a nested table.

### Update Of Fields

Specify the fields for UPDATE statement trigger upon necessary.

### ☒ Enable

You can choose whether to enable / disable the trigger constraint by checking / unchecking the box.

## Definition

Type in the definition for the trigger.

Example:

```
BEGIN
    add_job_history(:old.employee_id, :old.hire_date, sysdate,
                  :old.job_id, :old.department_id);
END;
```

## Advanced

### When Clause

Specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger. This condition must contain correlation names and cannot contain a query.

### Referencing Old

Specify correlation names. The default correlation names are OLD and NEW.

### Referencing New

Specify correlation names. The default correlation names are OLD and NEW.

### Follows

Specify the relative firing order of triggers of the same type.

**Note:** Support from Oracle 11g or later.

### Schema

Define the trigger on the current schema.

## Oracle Table Options

### Tablespace

Define a tablespace different from the default tablespace to create a table.

### Logging

Specify whether creation of a database object will be logged in the redo log file (LOGGING) or not (NOLOGGING).

### Compression

Specify whether to compress data segments to reduce disk use. It is valid only for heap-organized tables.

- *COMPRESS* - enables table compression.
- *COMPRESS FOR ALL OPERATIONS* - attempts to compress data during all DML operations on the table.
- *COMPRESS FOR DIRECT\_LOAD OPERATIONS* - attempts to compress data during direct-path INSERT operations when it is productive to do so.
- *NOCOMPRESS* - disables table compression.

### Cache

Indicate how blocks are stored in the buffer cache.

- *CACHE* - indicates that the blocks retrieved for this table are placed at the most recently used end of the least recently used (LRU) list in the buffer cache when a full table scan is performed.
- *NOCACHE* - indicates that the blocks retrieved for this table are placed at the least recently used end of the LRU list in the buffer cache when a full table scan is performed.

### ☒ Parallel With Degree

Specify the degree of parallelism, which is the number of parallel threads used in the parallel operation.

### ☒ Row Movement

With the option on, it allows the database to move a table row. It is possible for a row to move, for example, during table compression or an update operation on partitioned data.

### Physical Attributes

Refer to [Editing Physical Attributes/Default Storage Characteristics](#).

## Oracle External Tables

External tables access data in external sources as if it were in a table in the database. While creating external tables, you are actually creating metadata in the data dictionary that enables you to access external data.

Note that external tables are read only. No DML operations are possible and no index can be created.



## Table Designer for Oracle External Tables

**Table Designer** for External Tables allows you to create, edit table's fields, external properties and access parameters etc.



- [Managing External Table Fields](#)
- [Managing External Table External Properties](#)
- [Managing External Table Access Parameters](#)
- External Table SQL Preview

## Fields for Oracle External Tables

Table fields are managed on the **Fields** tab of the Table Designer. Just simply click a field for editing. A right-click displays the popup menu or by using the field toolbar, allowing you to create new and drop the selected field.

### Add Field

To add a field to the table

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click and select the  **Add Field** from the popup menu or click the  **Add Field** from the toolbar.
- Edit field properties.

To add a new field with modification as one of the existing fields

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Select field.
- Right-click and select the **Duplicate Field** from the popup menu.
- Edit field properties.



### Edit Field

To edit the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Simply click on the field to edit.

## Delete Field

To delete the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click and select the  **Delete Field** from the popup menu or click the  **Delete Field** from the toolbar.
- Confirm deleting in the dialog window.

## Setting Field Properties for Oracle External Tables

Name	Type	Length	Scale
▶ EMPLOYEE_ID	NUMBER	6	0
FIRST_NAME	VARCHAR2	20	0
LAST_NAME	VARCHAR2	25	0
EMAIL	VARCHAR2	25	0
PHONE_NUMBER	VARCHAR2	20	0

### Name

The Name is a descriptive identifier for a field that can be up to 30 characters by default (letters or numbers) including spaces. The names should be descriptive enough that anyone can easily identify them when viewing or editing records. For example, LastName, FirstName, StreetAddress, or HomePhone.

Use the **Name** edit box to set the field name. Note that the name of the field must be unique among all the field names in the table.

### Type

After you name a field, you choose a data type for the data to be contained in the field. When you choose a field's data type, you are deciding:

- What kind of values to allow in the field. You cannot store text in field with the **Numeric** data type.
- How much storage space Oracle is to set aside for the data in that field.
- What types of operations can be performed on the values in that field.

The **Type** dropdown list defines the type of the field data.

The following table shows all the built-in general-purpose data types.

Name	Description
CHAR	fixed-length character strings
NCHAR	fixed-length Unicode character data
VARCHAR2	variable-length character strings
VARCHAR	variable-length character strings
NVARCHAR2	variable-length Unicode character data
CLOB	database character set data
NCLOB	Unicode national character set data

LONG	variable-length character data containing up to 2 gigabytes of information
NUMBER	fixed and floating-point numbers
DATE	point-in-time values (dates and times)
INTERVAL DAY TO SECOND	period of time in terms of days, hours, minutes, and seconds
INTERVAL YEAR TO MONTH	stores a period of time using the YEAR and MONTH datetime fields
TIMESTAMP	point-in-time values (dates and times) (includes fractional seconds)
TIMESTAMP WITH TIME ZONE	TIMESTAMP with explicit time zone information
BLOB	unstructured binary data in the database
BFILE	unstructured binary data in operating-system files outside the database
RAW	can be indexed and is used for data that is not to be interpreted by Oracle Database
LONG RAW	cannot be indexed and is used for data that is not to be interpreted by Oracle Database
ROWID	the address (rowid) of every row in the database
CHARACTER	=CHAR <sup>1</sup>
CHARACTER VARYING	=VARCHAR2 <sup>1</sup>
CHAR VARYING	=VARCHAR2 <sup>1</sup>
NATIONAL CHARACTER	=NCHAR <sup>1</sup>
NATIONAL CHAR	=NCHAR <sup>1</sup>
NATIONAL CHARACTER VARYING	=NVARCHAR2 <sup>1</sup>
NATIONAL CHAR VARYING	=NVARCHAR2 <sup>1</sup>
NCHAR VARYING	=NVARCHAR2 <sup>1</sup>
NUMERIC	=NUMBER <sup>1</sup>
DECIMAL	=NUMBER <sup>1</sup>
INTEGER	=NUMBER(38) <sup>1</sup>
INT	=NUMBER(38) <sup>1</sup>
SMALLINT	=NUMBER(38) <sup>1</sup>
FLOAT	=FLOAT(126) <sup>1</sup>
DOUBLE PRECISION	=FLOAT(126) <sup>1</sup>
REAL	=FLOAT(63) <sup>1</sup>

**Note:** These are ANSI datatypes and datatypes from the IBM products SQL/DS and DB2. Oracle recognizes these datatypes and converts them to the equivalent Oracle datatype. [Click here](#) for detailed description on these datatypes.

## Length and Scale

Use the **Length** edit box to define the **precision** (total number of digits) of the field and use **Scale** edit box to define the **scale** (number of digits to the right of the decimal point) for **numeric** column.

**Note:** Be careful when shortening the field length as it might result in data loss.

How scale factors affect numeric data storage:

Input Data	Specified As	Stored As
7,456,123.89	NUMBER	7456123.89
7,456,123.89	NUMBER(*,1)	7456123.9
7,456,123.89	NUMBER(9)	7456124
7,456,123.89	NUMBER(9,2)	7456123.89
7,456,123.89	NUMBER(9,1)	7456123.9
7,456,123.89	NUMBER(6)	not accepted, exceeds precision
7,456,123.89	NUMBER(7,-2)	7456100

## Setting Other Field Properties for Oracle External Tables

For **INTERVAL DAY TO SECOND** data type:

### Leading Field Precision

Set the number of digits in the leading field. Accepted values are 0 to 9. The default is 2.

### Fractional Seconds Precision

Set the number of digits in the fractional part of the SECOND datetime field. Accepted values are 1 to 9.

For **INTERVAL YEAR TO MONTH** data type:

### Year Precision

Set the number of digits in the year. The default is 2.

For **TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE** data types:

### Fractional Seconds Precision

Set the number of digits in the fractional part of the SECOND datetime field. Accepted values are 1 to 9.

For **CHAR, VARCHAR2** data types:

### Unit

Set the unit either in BYTE or CHAR.

For **COMPLEX** data types:

### Object Schema

Set the object schema for the field.

### Object Type

Set the object type for the field.

## External Properties for Oracle External Tables

### Default Directory

Specify the default directory for the external table.

### Directory

Set the external directory.

### Location

Set the external source location.

### Access Driver

Specify the access driver for the external table. The default type for external tables is ORACLE\_LOADER.

### Reject Limit

Specify the limit on the number of errors that can occur during a query of the external data.

### ☒ Parallel With Degree

Check to enable parallel query on the data sources and specify the degree of parallel access.



## Access Parameters for Oracle External Tables

Describe the mapping of the external data to the Oracle Database data columns.

### ☒ **Using CLOB**

Check this option to get a CLOB data value of the returned query.

## Oracle Index Organized Tables

An index-organized table has a storage organization that is a variant of a primary B-tree. Data for an index-organized table is stored in a B-tree index structure in a primary key sorted manner. Each leaf block in the index structure stores both the key and nonkey columns.

Index-organized tables have full table functionality. They support features such as constraints, triggers etc with additional features such as key compression.

## Table Designer for Oracle Index Organized Tables

**Table Designer** for **Index Organized Tables** allows you to create, edit and drop table's fields, indexes, foreign keys, and much more.

Note that **Table Designer** for **Index Organized Tables** differs from **Normal Tables** only on the **Options** tab. Therefore, we will refer to **Table Designer for Normal Table(/Index Organized Table)** on the following similar chapters:

- [Managing Table Fields](#)
- [Managing Table Indexes](#)
- [Managing Table Foreign Keys](#)
- [Managing Table Uniques](#)
- [Managing Table Checks](#)
- [Managing Table Triggers](#)
- Managing Table Comment
- Table SQL Preview

### Options tab for Index Organized Table:

- [Managing Table Options](#)

## Options for Oracle Index Organized Tables

### Tablespace

Define a tablespace different from the default tablespace to create a table.

### Logging

Specify whether creation of a database object will be logged in the redo log file (LOGGING) or not (NOLOGGING).

#### ☒ Parallel With Degree

Specify the degree of parallelism, which is the number of parallel threads used in the parallel operation.

#### ☒ Row Movement

With the option on, it allows the database to move a table row. It is possible for a row to move, for example, during table compression or an update operation on partitioned data.

### Physical Attributes

Refer to [Editing Physical Attributes/Default Storage Characteristics](#).

### IOT Properties

#### ☒ Key Compress

Check this option to enable key compression. Upon necessary, you can also specify the prefix length (as the number of key columns), which identifies how the key columns are broken into a prefix and suffix entry.

### Mapping Table

Specify if there is a mapping table for the index-organized table. Note that a mapping table is required for creating bitmap indexes on an index-organized table.

### % Threshold

When an overflow segment is being used, it defines the maximum size of the portion of the row that is stored in the index block, as a percentage of block size.

#### ☒ Overflow Properties

Check to enable an overflow storage area.

**Note:** After saving the table, this option cannot be unchecked.

**Overflow Column**

Specify the column to be put in a separate overflow data segment.

**Overflow Tablespace**

Specify the tablespace in which the overflow segment to be stored.

**Overflow Logging**

Specify whether creation of a database object will be logged in the redo log file (LOGGING) or not (NOLOGGING).


**Overflow Physical Attributes**

Refer to [Editing Physical Attributes/Default Storage Characteristics](#).





## Oracle Views


Views are useful for allowing users to access a set of relations (tables) as if it were a single table, and limiting their access to just that. Views can also be used to restrict access to rows (a subset of a particular table).

Just simply click  to open an object pane for **View**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit, open and delete the selected view.

### Create View

To create a new view

- Select anywhere on the object pane.
- Click the  **New View** from the object pane toolbar.  
or
- Right-click and select  **New View** from the popup menu.
- Edit view properties on the appropriate tabs of the View Designer.

**Hint:** To create new view you can also right-click the Views node of the navigation pane and select the  **New View** from the popup menu.

To create a new view with the same properties as one of the existing views has (using drag and drop method)




**Apply to:** current schema {same connection}

- Select the view(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen view(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created view(s) will be named as "viewname\_**copy**".




**Apply to:** different schema {same connection}  
different schema {different connection} (Data Transfer tool will be activated)

- Select the view(s) for copying in the object pane.
- Drag and drop the chosen view(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel

To create a new view with modification as one of the existing views

- Select the view for modifying in the navigation pane/object pane.
- Right-click and select the  **Design View** from the popup menu.  
or
- Click the  **Design View** from the object pane toolbar.
- Modify view properties on the appropriate tabs of the View Designer.
- Click  **Save As**.

To create a new view with loading from a SQL file

- Select anywhere on the object pane.
- Click the  **New View** from the object pane toolbar.  
or
- Right-click and select  **New View** from the popup menu.
- Click  **Load**.

## Create View Shortcut



To create a view shortcut

- Select the view for editing in the navigation pane/object pane.
- Right-click and select **Create Open View Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.

**Note:** This option is used to provide a convenient way for you to open your view for entering data directly (Grid View/Form View) without activating the main Navicat.

## Edit View

To edit the existing view (manage its SQL definition etc)



- Select the view for editing in the navigation pane/object pane.
- Right-click and select the  **Design View** from the popup menu.  
or
- Click the  **Design View** from the object pane toolbar.
- Edit view properties on the appropriate tabs of the View Designer.

To change the name of the view

- Select the view for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.

## Open View

To open a view (manage view data)

- Select the view for opening in the navigation pane/object pane.
- Right-click and select the  **Open View** from the popup menu or simply double-click the view.  
or
- Click the  **Open View** from the object pane toolbar.

## Maintain View



To maintain a view

- Select the view for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Compile



## Delete View

To delete a view

- Select the view for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete View** from the popup menu.  
or
- Click the  **Delete View** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve View Information

To achieve a view information

- Select the view in the navigation pane/object pane.
- Right-click the selected view and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle View Designer

**View Designer** is the basic Navicat tool for working with views. It allows you to create new view and edit the existing view definition (view name and the SELECT statement it implements).

- [Working with View Builder](#)
- [Editing View SQL Definition](#)
- [Setting Advanced View Properties](#)
- Editing View Comment
- View SQL Preview
- [View Preview](#)
- [View Explain](#)

## **Working with Oracle View Builder (Available only in Full Version)**

**View Builder** allows you to build views visually. It allows you to create and edit views without knowledge of SQL. See Query Builder for details.

## Editing Oracle View SQL Definition

The **Definition** tab allows you to edit the view definition as SQL statement (SELECT statement it implements).

Example:

```
SELECT
  HR.EMPLOYEES.EMPLOYEE_ID
FROM
  HR.EMPLOYEES
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.

## Setting Advanced Oracle View Properties

The **Advanced** tab allows you to restrict the defining query of the view.

### ☒ **Restrict Query**

Specify the name. If you omit this identifier, then Oracle automatically assigns a name of the form SYS\_Cn, where n is an integer that makes the constraint name unique within the database.

### **Read Only**

Indicate that the table or view cannot be updated.


### **Check option**

Indicate that Oracle Database prohibits any changes to the table or view that would produce rows that are not included in the subquery.

### ☒ **Force On Create**


Check this option if you want to create the view regardless of whether the base tables of the view or the referenced object types exist or the owner of the schema containing the view has privileges on them.

## Oracle View Preview

To preview the result of the view, click  **Preview** on the toolbar. If the query statement is correct, the **Result** and **Message** tabs will be opened.



The **Result** tab displays the data of the view as a grid and the **Message** tab displays the message log.

## Oracle View Explain

To show the Explain Plan of the query, click  **Explain** on the toolbar. If the query statement is correct, the **Explain** tab opens with the columns in the PLAN\_TABLE.

Column	Description
Operation	Name of the internal operation performed in this step.
Object	Name of the table or index.
Optimizer	Current mode of the optimizer.
Cost	Cost of the operation as estimated by the optimizer's query approach. Cost is not determined for table access operations. The value of this column does not have any particular unit of measurement; it is merely a weighted value used to compare costs of execution plans. The value of this column is a function of the CPU_COST and IO_COST columns.
Cardinality	Estimate by the query optimization approach of the number of rows accessed by the operation.
Bytes	Estimate by the query optimization approach of the number of bytes accessed by the operation.
Partition Start	Start partition of a range of accessed partitions.
Partition ID	Step that has computed the pair of values of the PARTITION_START and PARTITION_STOP columns.
Access Predicates	Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
Filter Predicates	Predicates used to filter rows before producing them.

## Oracle View Viewer

**View Viewer** displays the view data as a grid. Data can be displayed in three modes:  **Grid View**,  **Form View** and **Text/Blob/BFile View**. See Data View for details.

The toolbars of View Viewer provides the following functions for managing data:

- **Commit**

Make permanent all changes performed in the transaction.

**Hint:** The Commit button is visible only when **Auto Commit** is disabled under Option Settings.

- **Rollback**

Undo work done in the current transaction.

**Hint:** The Rollback button is visible only when **Auto Commit** is disabled under Option Settings.

- **Export Data**

Export data to MS Word, MS Excel, MS Access, TXT, DBF, HTML, SQL, RTF and more.

- **Filter Data**

Allow you to filter records by creating and applying filter criteria for the data grid.

- **Edit TEXT/BLOB/BFile**

Allow you to view and edit the content of TEXT, BLOB and BFile fields.



EMP\_DETAILS\_VIEW @HR (Basic Connection)

File Edit View Window Help

Commit Rollback Export Wizard Filter Wizard Grid View Form View


EMPLOYEE_ID	JOB_ID	MANAGER_ID	DEPARTMENT_ID
100	AD_PRES	(Null)	
101	AD_VP		100
102	AD_VP		100
103	IT_PROG		102
104	IT_PROG		103
105	IT_PROG		103
106	IT_PROG		103
107	IT_PROG		103
108	FI_MGR		101
109	FI_ACCOUNT		108
110	FI_ACCOUNT		108
111	FI_ACCOUNT		108
112	FI_ACCOUNT		108
113	FI_ACCOUNT		108
114	PU_MAN		100
115	PU_CLERK		114
116	PU_CLERK		114
117	PU_CLERK		114
118	PU_CLERK		114
119	PU_CLERK		114
120	ST_MAN		100

SELECT \* FROM (SELECT "NAVICAT\_TABLE".\*, ROWNUM "NA Record 1 of 106 in Page 1

## Oracle Functions/Procedures



A procedure or function is a schema object that consists of a set of SQL statements and other PL/SQL constructs, grouped together, stored in the database, and run as a unit to solve a specific problem or perform a set of related tasks.


Procedures and functions are identical except that functions always return a single value to the caller, while procedures do not.

Just simply click  to open an object pane for **Function**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected function/procedure.

### Create Function/Procedure

To create a new function/procedure

- Select anywhere on the object pane.
- Click the  **New Function** from the object pane toolbar.  
or
- Right-click and select  **New Function** from the popup menu.
- Edit function/procedure properties on the appropriate tabs of the Function/Procedure Designer.

**Hint:** To create new function/procedure you can also right-click the Function node of the navigation pane and select the  **New Function** from the popup menu.

To create a new function/procedure with the same properties as one of the existing function/procedure has (using drag and drop method)

**Apply to:** current schema {same connection}

- Select the function/procedure(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen function/procedure(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created function/procedure(s) will be named as "function/procedurename\_**copy**".



**Apply to:** different schema {same connection}

different schema {different connection} (Data Transfer tool will be activated)

- Select the function/procedure(s) for copying in the object pane.
- Drag and drop the chosen function/procedure(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel



## Edit Function/Procedure

To edit the existing function/procedure


- Select the function/procedure for editing in the navigation pane/object pane.
- Right-click and select the  **Design Function** from the popup menu or simply double-click the function/ procedure.  
or
- Click the  **Design Function** from the object pane toolbar.
- Edit function/procedure properties on the appropriate tabs of the Function/Procedure Designer.

## Run Function/Procedure

To run a function/procedure in the navigation pane/object pane

- Select the function/procedure for executing in the navigation pane/object pane.
- Click the  **Execute Function** from the object pane toolbar.  
or
- Right-click and select  **Execute Function** from the popup menu.
- View the returned data on the DBMS Output tab.

To run a function/procedure in the Function/Procedure Designer


- Create a new function/procedure or open the existing function/procedure.
- Click  **Run**.
- View the returned data on the DBMS Output tab.

## Debug Function/Procedure

To debug a function/procedure in the object pane

- Select the function/procedure for debugging in the object pane.
- Right-click and select the **Debug Function** from the popup menu.
- Debug the function/procedure in the Debugger.

To debug a function/procedure in the Function/Procedure Designer

- Create a new function/procedure or open the existing function/procedure.
- Click  **Debug**.
- Debug the function/procedure in the Debugger.



## Maintain Function/Procedure

To maintain a function/procedure

- Select the function/procedure for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Compile
  - Compile for Debug

## Delete Function/Procedure

To delete a function/procedure


- Select the function/procedure for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Function** from the popup menu.  
or
- Click the  **Delete Function** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Function/Procedure Information

To achieve a function/procedure information

- Select the function/procedure in the navigation pane/object pane.
- Right-click the selected function/procedure and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Function Wizard

Click the  **New Function** from the object pane toolbar. The **Function Wizard** will pop up and it allows you to create a procedure/function easily.

- [Setting Routine Type](#)
- [Setting Parameters for Procedure/Function](#)
- [Setting Return Type for Function](#)

You are allowed not to show the **Function Wizard** when create new procedure/function.

**Hint:** Once uncheck the **Show wizard next time**, you can go to Options to enable it.

## Setting Oracle Routine Type

Specify the **Name** of the routine.

Select the type of the routine: **Procedure** or **Function**

## Setting Parameters for Oracle Procedure/Function

Define the parameter(s) of the procedure/function. Set the parameter **Name**, **Type**, **Mode** and **Default Value** under corresponding columns.



## Setting Return Type for Oracle Function

Select the **Return Type** from the list.

**Note:** Only function supports return type.

## Oracle Function/Procedure Designer

**Function/Procedure Designer** allows you to edit the existing function/procedure definition and more.







- [Editing Function/Procedure Definition](#)
- Function/Procedure SQL Preview
- [Viewing Function/Procedure Result](#)
- [Debugging Function/Procedure](#)

## Editing Oracle Function/Procedure Definition

Edit the function/procedure definition under the **Definition** tab.

The **Code Outline** window displays information about the function/procedure including parameter, code body, etc. To show the **Code Outline** window, simply choose View -> **Code Outline**.

**Note:** Available only in Full Version.


	Refresh the code outline.
	Show the detail view of the code outline.
	Turn mouse over highlight on or off.
	Expand the selected item.
	Collapse the selected item.
	Toggle sorting by position.

The SQL statements for creating procedures are CREATE PROCEDURE. In practice, it is best to use a CREATE OR REPLACE statement. The general form of these statements follows.

```
CREATE OR REPLACE
PROCEDURE "" AS
BEGIN
    -- routine body goes here, e.g.
    -- DBMS_OUTPUT.PUT_LINE('Navicat for Oracle');
END;
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.


## Viewing Oracle Function/Procedure Result

To run the function/procedure click  **Run** on the toolbar. If the SQL statement is correct, the statement will be executed and, if the statement is supposed to return data, the **Message** and **DBMS Output** tabs open with the message log and data returned by the function/procedure. If an error occurs while executing the function/procedure, execution stops, the appropriate error message is displayed.

If the function/procedure requires input parameter, the **Input Parameters** box will popup.

## Debugging Oracle Function/Procedure (Available only in Full Version)

To debug the function/procedure click  **Debug** on the toolbar to launch the [Oracle Debugger](#).

You can add/remove breakpoints for debugging by clicking  in the grey area beside each statement.

Enter the Input Parameters if necessary.


```

1  CREATE OR REPLACE
2  FUNCTION str2tbl( p_str IN VARCHAR2 ) RETURN
   charTableType
3  AS
4      l_str      LONG DEFAULT p_str || ',';
5      l_n        NUMBER;
6      l_data     charTableType := charTableType();
7  BEGIN
8      LOOP
9          l_n := instr( l_str, ',' );
10         exit when (nvl(l_n,0) = 0);
11         l_data.extend;
12         l_data( l_data.count ) := ltrim(rtrim(
            substr(l_str,1,l_n-1));
13         l_str := substr( l_str, l_n+1 );
14     END LOOP;
15     RETURN l_data;
16 END;
17

```



## Oracle Database Links

Database link is a named schema object that describes a path from one database to another and are implicitly used when a reference is made to a global object name in a distributed database. After you have created a database link, you can use it to refer to tables and views on the other database.

Just simply click  -> **Database Link** to open an object pane for **Database Link**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new and delete the selected database link.



### Create Database Link

To create a new database link

- Select anywhere on the object pane.
- Click the  **New Database Link** from the object pane toolbar.  
or
- Right-click and select  **New Database Link** from the popup menu.
- Edit database link properties on the appropriate tabs of the Database Link Designer.

### Delete Database Link

To delete a database link

- Select the database link for deleting in the object pane.
- Right-click and select the  **Delete Database Link** from the popup menu.  
or
- Click the  **Delete Database Link** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Database Link Information

To achieve a database link information

- Select the database link in the object pane.
- Right-click the selected database link and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Database Link Designer

**Database Link Designer** is the basic Navicat tool for working with database links. It allows you to create new database link.

- [Editing Database Link General](#)
- Database Link SQL Preview



## Edit Oracle Database Link General

### Service Name

Specify the service name of a remote database.

### User Name

The user name used to connect to the remote database using a fixed user database link.

### Password

The password for connecting to the remote database.

### ☒ Current user


With this option checked, a current user database link is created. The current user must be a global user with a valid account on the remote database.

### ☒ Shared

Fill in **Authentication username** and **Authentication password** when Shared option is enabled.



## Oracle Indexes

Index provides a faster access path to table data. It is created using one or more columns of a table to speed SQL statement execution on that table.




Just simply click  -> **Index** to open an object pane for **Index**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected Index.

### Create Index

To create a new index



- Select anywhere on the object pane.
- Click the  **New Index** from the object pane toolbar.  
or
- Right-click and select  **New Index** from the popup menu.
- Edit index properties on the appropriate tabs of the Index Designer.

To create a new index with modification as one of the existing index

- Select the index for modifying in the object pane.
- Right-click and select the  **Design Index** from the popup menu or simply double-click the index.  
or
- Click the  **Design Index** from the object pane toolbar.
- Modify index properties on the appropriate tabs of the Index Designer.
- Click  **Save As**.

### Edit Index

To edit the existing index (manage its properties etc)

- Select the index for editing in the object pane.
- Right-click and select the  **Design Index** from the popup menu or simply double-click the index.  
or
- Click the  **Design Index** from the object pane toolbar.
- Edit index properties on the appropriate tabs of the Index Designer.

To change the name of the index

- Select the index for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.



## Maintain Index

To maintain an index

- Select the index for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Rebuild
  - Make Unusable
  - Coalesce
  - Compute Statistics
  - Monitoring Usage
  - No Monitoring Usage

## Delete Index

To delete an index

- Select the index for deleting in the object pane.
- Right-click and select the  **Delete Index** from the popup menu.  
or
- Click the  **Delete Index** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Index Information

To achieve an index information

- Select the index in the object pane.
- Right-click the selected index and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Index Designer

**Index Designer** is the basic Navicat tool for working with indexes. It allows you to create new index and edit the existing index properties.

- [Editing Index General](#)
- [Editing Advanced Index Properties](#)
- Index SQL Preview

## Editing Oracle Index General

### Type

The types of the index.

#### Normal

Normal Index (A B-tree index)

#### Unique

Unique Index (No two rows of a table have duplicate values in the key columns)

#### Bitmap

Bitmap Index (A bitmap for each key value)

#### Domain

Domain Index (Instances of an application-specific index)

#### Cluster

Cluster Index

## Oracle Normal Index

A normal index does not impose restrictions on the column values.

### Type

Choose between **Normal**, Unique, Bitmap, Domain and Cluster.

### Table schema

The schema that contains the index.

### Table name

The table name.

### Fields

Use the **Name** dropdown list to select the field name and **Descend** dropdown list to define the order of the index (ASC or DESC).

## Oracle Unique Index

A unique index indicates that no two rows of a table have duplicate values in the key columns.

### Type

Choose between Normal, **Unique**, Bitmap, Domain and Cluster.

### Table schema

The schema that contains the index.

### Table name

The table name.

### Fields

Use the **Name** dropdown list to select the field name and **Descend** dropdown list to define the order of the index (ASC or DESC).

## Oracle Bitmap Index

A bitmap index created with a bitmap for each distinct key, rather than indexing each row separately. Bitmap indexes store the rowids associated with a key value as a bitmap. Each bit in the bitmap corresponds to a possible rowid.

### Type

Choose between Normal, Unique, **Bitmap**, Domain and Cluster.

### Table schema

The schema that contains the index.

### Table name

The table name.

### Fields

Use the **Name** dropdown list to select the field name and **Descend** dropdown list to define the order of the index (ASC or DESC).

### ☒ Bitmap Join Index

In addition to a bitmap index on a single table, you can create a bitmap join index, which is a bitmap index for the join of two or more tables. A bitmap join index is a space efficient way of reducing the volume of data that must be joined by performing restrictions in advance.

#### Fields

Use the **Schema**, **Table** and **Name** dropdown lists to select the schema, table and field name and **Descend** dropdown list to define the order of the index (ASC or DESC).

#### Bitmap Join

Use the **InnerSchema**, **InnerTable**, **InnerField**, **OuterSchema**, **OuterTable** and **OuterField** dropdown lists to select joined schemas, tables and fields respectively.



## Oracle Domain Index

A domain index is an index designed for a specialized domain, such as spatial or image processing. Users can build a domain index of a given type after the designer creates the indextype.

### Type

Choose between Normal, Unique, Bitmap, **Domain** and Cluster.

### Table schema

The schema that contains the index.

### Table name

The table name.

### Column

The column which the index is based.

### Index Type

#### Schema

The schema of the indextype.

#### Type

Select the created or built-in indextypes.

#### Parameters

Information about the path table and about the secondary indexes corresponding to the components of XMLIndex. The maximum length of the parameter string is 1000 characters.

## Oracle Cluster Index

A cluster index is an index designed for a cluster.

### Type

Choose between Normal, Unique, Bitmap, Domain and **Cluster**.

### Table schema

The schema that contains the index.

### Cluster name

The name of the cluster.

## Editing Advanced Oracle Index Properties

### ☒ **Unusable**

An unusable index must be rebuilt, or dropped and re-created, before it can be used.

### **Tablespace**

The name of the tablespace to hold the index.

### ☒ **Compress**

To enable key compression, which eliminates repeated occurrence of key column values and may substantially reduce storage.

**Note:** No compression for Bitmap Indexes.

### ☒ **Parallel**

The creation of the index will be parallelized.

### ☒ **Reverse**

To store the bytes of the index block in reverse order, excluding the rowid.

## Logging

### **Logging**

The creation of the index will be logged in the redo log file.

### **No Logging**

The creation of the index will be not logged in the redo log file.

## Visibility

### **Visible**

Specify the index is visible to the optimizer.

### **Invisible**

Specify the index is invisible to the optimizer.

## Create / Rebuild Option

### ☒ **Online**

To indicate that DML operations on the table will be allowed during creation of the index.

## ☒ **No Sort**


To indicate to the database that the rows are already stored in the database in ascending order, so that Oracle Database does not have to sort the rows when creating the index.

### **Physical Attributes**

Set the physical attributes of an index.



## Oracle Java

Java is an object-oriented programming language efficient for application-level programs. You can write and load applications within the database.




Just simply click -> **Java** to open an object pane for **Java**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected Java.

## Create Java

To create a new Java



- Select anywhere on the object pane.
- Click the  **New Java** from the object pane toolbar.  
or
- Right-click and select  **New Java** from the popup menu.
- Edit Java properties on the appropriate tabs of the Java Designer.

To create a new Java with modification as one of the existing Java

- Select the Java for modifying in the object pane.
- Right-click and select the  **Design Java** from the popup menu or simply double-click the Java.  
or
- Click the  **Design Java** from the object pane toolbar.
- Modify Java properties on the appropriate tabs of the Java Designer.
- Click  **Save As**.

## Edit Java

To edit the existing Java(manage its general, advanced etc)

- Select the Java for editing in the object pane.
- Right-click and select the  **Design Java** from the popup menu or simply double-click the Java.  
or
- Click the  **Design Java** from the object pane toolbar.
- Edit Java properties on the appropriate tabs of the Java Designer.



## Maintain Java

To maintain a Java

- Select the Java for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Compile or Resolve
  - Set AuthID Current User
  - Set AuthID Definer

## Delete Java

To delete a Java

- Select the Java for deleting in the object pane.
- Right-click and select the  **Delete Java** from the popup menu.  
or
- Click the  **Delete Java** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Java Information

To achieve a Java information

- Select the Java in the object pane.
- Right-click the selected Java and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Java Designer

**Java Designer** is the basic Navicat tool for working with Java. It allows you to create new Java and edit the existing Java properties.

- [Editing Java General](#)
- [Setting Advanced Java Properties](#)
- Java SQL Preview

## Editing Oracle Java General

You can create a Java source, class, or resource using the Java Designer.

- [Java Source](#)
- [Java Class](#)
- [Java Resource](#)



## Oracle Java Source

### Type

Choose between **Java Source**, Java Class and Java Resource.

### Create from

#### BFile

Select the **Directory** and type the **Server Filename**.

#### Load from file

Browse the **File Path** of Java source file.

#### Plain source

Type the source code in the **Source** box.

## Oracle Java Class

### Type

Choose between Java Source, **Java Class** and Java Resource.

### Create from

#### BFile

Select the **Directory** and type the **Server Filename**.

#### Load from file

Browse the **File Path** of the Java class file.

## Oracle Java Resource

### Type

Choose between Java Source, Java Class and **Java Resource**.

### Create from

#### BFile

Select the **Directory** and type the **Server Filename**.

#### Load from file

Browse the **File Path** of the Java resource file.

## Setting Advanced Oracle Java Properties

### Invoker Rights

Select **CURRENT\_USER** to indicate that the methods of the class execute with the privileges of CURRENT\_USER or **DEFINER** indicates that the methods of the class execute with the privileges of the owner of the schema in which the class resides, and that external names resolve in the schema where the class resides.

### Resolver

Specify a mapping of the fully qualified Java name to a Java schema object.

#### ☒ **Compile or Resolve**


Check this to specify that Oracle Database should attempt to resolve the Java schema object that is created if this statement succeeds.

#### ☒ **No Force**

Check this to roll back the results of the CREATE command of Java if you have enabled Compile or Resolve and the resolution or compilation fails. If you do not specify this option, then Oracle Database takes no action if the resolution or compilation fails, and the created schema object remains.



## Oracle Materialized Views

Materialized view is a schema object that can be used to summarize, compute, replicate, and distribute data.




Just simply click -> **Materialized View** to open an object pane for **Materialized View**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected materialized view.

### Create Materialized View




To create a new materialized view

- Select anywhere on the object pane.
- Click the  **New Materialized View** from the object pane toolbar.  
or
- Right-click and select  **New Materialized View** from the popup menu.
- Edit materialized view properties on the appropriate tabs of the Materialized View Designer.

To create a new materialized view with modification as one of the existing materialized view

- Select the materialized view for modifying in the object pane.
- Right-click and select the  **Design Materialized View** from the popup menu.  
or
- Click the  **Design Materialized View** from the object pane toolbar.
- Modify materialized view properties on the appropriate tabs of the Materialized View Designer.
- Click  **Save As**.

To create a new materialized view with loading from a SQL file

- Select anywhere on the object pane.
- Click the  **New Materialized View** from the object pane toolbar.  
or
- Right-click and select  **New Materialized View** from the popup menu.
- Click  **Load**.

## Create Materialized View Shortcut



To create a materialized view shortcut

- Select the materialized view for editing in the object pane.
- Right-click and select **Create Open Materialized View Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.

**Note:** This option is used to provide a convenient way for you to open your materialized view for editing data directly (Grid View/Form View) without activating the main Navicat.



## Edit Materialized View

To edit the existing materialized view (manage its properties etc)

- Select the materialized view for editing in the object pane.
- Right-click and select the  **Design Materialized View** from the popup menu.  
or
- Click the  **Design Materialized View** from the object pane toolbar.
- Edit materialized view properties on the appropriate tabs of the Materialized View Designer.

## Open Materialized View

To open a materialized view (manage materialized view data)

- Select the materialized view for opening in the object pane.
- Right-click and select the  **Open Materialized View** from the popup menu or simply double-click the materialized view.  
or
- Click the  **Open Materialized View** from the object pane toolbar.



## Maintain Materialized View

To maintain a materialized view

- Select the materialized view for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Enable Row Movement
  - Shrink
  - Compile
  - Force Refresh

## Delete Materialized View

To delete a materialized view

- Select the materialized view for deleting in the object pane.
- Right-click and select the  **Delete Materialized View** from the popup menu.  
or
- Click the  **Delete Materialized View** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Materialized View Information

To achieve a materialized view information

- Select the materialized view in the object pane.
- Right-click the selected materialized view and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Materialized View Designer

**Materialized View Designer** is the basic Navicat tool for working with materialized views. It allows you to create new materialized view and edit the existing materialized view properties.

- [Working with Materialized View Builder](#)
- [Editing Materialized View SQL Definition](#)
- [Setting Advanced Materialized View Properties](#)
- Editing Materialized View Comment
- Materialized View SQL Preview



## **Working with Oracle Materialized View Builder (Available only in Full Version)**

**View Builder** allows you to build views visually. It allows you to create and edit materialized views without knowledge of SQL. See Query Builder for details.

## Editing Oracle Materialized View SQL Definition

The **Definition** tab allows you to edit the view definition as SQL statement (SELECT statement it implements).

Example:

```
SELECT
  HR.EMPLOYEES.EMPLOYEE_ID
FROM
  HR.EMPLOYEES
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.

## Setting Advanced Oracle Materialized View Properties

### Refresh Options

#### When

##### **Demand**

The materialized view will be refreshed on demand by calling one of the three DBMS\_MVIEW refresh procedures.

##### **Commit**

A fast refresh is to occur whenever the database commits a transaction that operates on a master table of the materialized view.

##### **Automatic**

The database automatically refresh the materialized view with the automatic refresh time.

##### **Never**

The materialized view will not be refreshed with any Oracle Database refresh mechanism or packaged procedure.

#### Method

##### **FORCE**

When a refresh occurs, Oracle Database will perform a fast refresh if one is possible or a complete refresh if fast refresh is not possible.

##### **FAST**

A incremental refresh method, which performs the refresh according to the changes that have occurred to the master tables.

##### **COMPLETE**

A complete refresh method, which is implemented by executing the defining query of the materialized view.

#### Start On

A datetime expression for the first automatic refresh time.

#### Next

A datetime expression for calculating the interval between automatic refreshes.

**Note:** To edit the datetime, just simply click ... and choose / enter the desired data.

## Type

### Primary Key

A primary key materialized view. This is the default.

### Row ID

A rowid materialized view.

## Rollback Segment

### ☒ Master

The remote rollback segment is used at the remote master site for the individual materialized view.

### ☒ Local

The remote rollback segment is used for the local refresh group that contains the materialized view. This is the default.

## Constraints

### Enforced

Oracle Database use enforced constraints during the refresh operation.

### Trusted

Oracle Database use dimension and constraint information that has been declared trustworthy by the database administrator but that has not been validated by the database.

## Create Options

### ☒ No Index

Check this to suppress the creation of the default index.

## Build Type

### Immediate

The materialized view is to be populated immediately. This is the default.

## **Deferred**

The materialized view is to be populated by the next refresh operation.

## **Prebuilt**

To register an existing table as a preinitialized materialized view.

## **Materialized View Options**

### **Reduced Precision**

#### **With**

To authorize the loss of precision that will result if the precision of the table or materialized view columns do not exactly match the precision returned by subquery.

#### **Without**

To require that the precision of the table or materialized view columns match exactly the precision returned by subquery, or the create operation will fail. This is the default.

### **Compress**

Data segments are compressed to reduce disk and memory use.

### **Parallel**

Choose **NOPARALLEL** for serial execution or **PARALLEL** if you want Oracle to select a degree of parallelism equal to the number of CPUs available on all participating instances times the value of the PARALLEL\_THREADS\_PER\_CPU initialization parameter.

#### **With Degree**

Set the default degree of parallelism for queries and DML on the materialized view after creation.

### **Logging**

Choose **LOGGING** for logging the creation of Materialized view in the redo log file.

Choose **NOLOGGING** for no logging.

### **Tablespace**

Choose the tablespace in which the materialized view is to be created.

### **Physical Attributes**

Set the physical attributes of the materialized view.

## ☒ **Enable Cache**

The blocks retrieved for the table are placed at the most recently used end of the least recently used (LRU) list in the buffer cache when a full table scan is performed.

## ☒ **For Update**

Check this to allow a subquery, primary key, object, or rowid materialized view to be updated. When used in conjunction with Advanced Replication, these updates will be propagated to the master.

## ☒ **Enable Query Rewrite**

The materialized view is used for query rewrite.

## **Using Index Clause**


### **Tablespace**

Choose the tablespace of the index.

### **Physical Attributes**


Set the physical attributes for the default index Oracle Database uses to maintain the materialized view data.

## Oracle Materialized View Preview

To preview the result of the view, click  **Preview** on the toolbar. If the query statement is correct, the **Result** and **Message** tabs will be opened.

The **Result** tab displays the data of the view as a grid and the **Message** tab displays the message log.

## Oracle Materialized View Explain


To show the Explain Plan of the query, click  **Explain** on the toolbar. If the query statement is correct, the **Explain** tab opens with the columns in the PLAN\_TABLE.

Column	Description
Operation	Name of the internal operation performed in this step. In the first row generated for a statement.
Object	Name of the table or index.
Optimizer	Current mode of the optimizer.
Cost	Cost of the operation as estimated by the optimizer's query approach. Cost is not determined for table access operations. The value of this column does not have any particular unit of measurement; it is merely a weighted value used to compare costs of execution plans. The value of this column is a function of the CPU_COST and IO_COST columns.
Cardinality	Estimate by the query optimization approach of the number of rows accessed by the operation.
Bytes	Estimate by the query optimization approach of the number of bytes accessed by the operation.
Partition Start	Start partition of a range of accessed partitions.
Partition ID	Step that has computed the pair of values of the PARTITION_START and PARTITION_STOP columns.
Access Predicates	Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
Filter Predicates	Predicates used to filter rows before producing them.





## Oracle Materialized View Logs

Materialized view log is a schema object that records changes to a master table's data so that a [Materialized View](#) defined on the master table can be refreshed incrementally.




Just simply click -> **Materialized View Log** to open an object pane for **Materialized View Log**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected materialized view log.

### Create Materialized View Log

To create a new materialized view log



- Select anywhere on the object pane.
- Click the  **New Materialized View Log** from the object pane toolbar.  
or
- Right-click and select  **New Materialized View Log** from the popup menu.
- Edit materialized view log properties on the appropriate tabs of the Materialized View Log Designer.

To create a new materialized view log with modification as one of the existing materialized view log

- Select the materialized view log for modifying in the object pane.
- Right-click and select the  **Design Materialized View Log** from the popup menu or simply double-click the materialized view log.  
or
- Click the  **Design Materialized View Log** from the object pane toolbar.
- Modify materialized view log properties on the appropriate tabs of the Materialized View Log Designer.
- Click  **Save As**.



## Edit Materialized View Log

To edit the existing materialized view log (manage its general, advance, etc)

- Select the materialized view log for editing in the object pane.
- Right-click and select the  **Design Materialized View Log** from the popup menu or simply double-click the materialized view log.  
or
- Click the  **Design Materialized View Log** from the object pane toolbar.
- Edit materialized view log properties on the appropriate tabs of the Materialized View Log Designer.

## Open Materialized View Log Table

To open a materialized view log table (manage materialized view log data)

- Select the materialized view log table for opening in the object pane.
- Right-click and select the  **Open Materialized View Log Table** from the popup menu  
or
- Click the  **Open Materialized View Log Table** from the object pane toolbar.



## Maintain Materialized View Log

To maintain a materialized view log

- Select the materialized view log table for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Enable Row Movement
  - Disable Row Movement
  - Shrink Space

## Delete Materialized View Log

To delete a materialized view log

- Select the materialized view log for deleting in the object pane.
- Right-click and select the  **Delete Materialized View Log** from the popup menu.  
or
- Click the  **Delete Materialized View Log** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Materialized View Log Information

To achieve a materialized view log information

- Select the materialized view log in the object pane.
- Right-click the selected materialized view log and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Materialized View Log Designer

**Materialized View Log Designer** is the basic Navicat tool for working with materialized view logs. It allows you to create new materialized view log and edit the existing materialized view log definition.

- [Editing Materialized View Log General](#)
- Materialized View Log SQL Preview

## Editing Oracle Materialized View Log General

Edit the materialized view log general properties under the **General** tab.

### Table

The table of the materialized view log.

### Tablespace

The tablespace of the materialized view log.

### Logging

To specify either **LOGGING** or **NOLOGGING** to establish the logging characteristics for the materialized view log.

### Cache

#### **CACHE**

The blocks retrieved for this log are placed at the most recently used end of the least recently used (LRU) list in the buffer cache when a full table scan is performed.

#### **NOCACHE**

The blocks are placed at the least recently used end of the LRU list. This is the default.

### New Values

#### **INCLUDING**

To save both new and old values in the log.

#### **EXCLUDING**

To disable the recording of new values in the log.

#### ☒ **Parallel With Degree**

To determine the number of parallel threads used in the parallel operation.

### [Physical Attributes](#)

Set the physical attributes of a materialized view log.

## With

### ☒ **Object ID**

The system-generated or user-defined object identifier of every modified row should be recorded in the materialized view log.

### ☒ **Primary Key**

The primary key of all rows changed should be recorded in the materialized view log.

### ☒ **Row ID**

The rowid of all rows changed should be recorded in the materialized view log.

### ☒ **Sequence**

A sequence value providing additional ordering information should be recorded in the materialized view log.

## **Fields**

Choose the fields whose values you want to be recorded in the materialized view log for all rows that are changed.



## Oracle Packages

Packages are encapsulated collections of related procedures, stored functions, and other program objects stored together in the database. Package bodies, specified subsequently, defines these objects. An package consists of two parts: a specification and a body.

Just simply click -> **Package** to open an object pane for **Package**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected package.

### Create Package (Package Specification)

To create a new package

- Select anywhere on the object pane.
- Click the **New Package** from the object pane toolbar.  
or
- Right-click and select **New Package** from the popup menu.
- Edit package properties on the appropriate tabs of the Package Designer.

### Edit Package

To edit the existing package (manage its definition etc)

- Select the package for editing in the object pane.
- Right-click and select the **Design Package** from the popup menu or simply double-click the package.  
or
- Click the **Design Package** from the object pane toolbar.
- Edit package properties on the appropriate tabs of the Package Designer.


### Delete Package

To delete a package

- Select the package for deleting in the object pane.
- Right-click and select the **Delete Package** from the popup menu.  
or
- Click the **Delete Package** from the object pane toolbar.
- Confirm deleting in the dialog window.


## Create Package Body (Package Body)

To create a new package body

- Select the package for modifying in the object pane.
- Right-click and select  **New Package Body** from the popup menu.
- Edit package body properties on the appropriate tabs of the Package Body Designer.


## Edit Package Body

To edit the existing package body (manage its definition etc)

- Select the package for editing in the object pane.
- Right-click and select the  **Design Package Body** from the popup menu.
- Edit package body properties on the appropriate tabs of the Package Body Designer.



## Delete Package Body

To delete a package body


- Select the package for deleting in the object pane.
- Right-click and select the  **Delete Package Body** from the popup menu.
- Confirm deleting in the dialog window.

## Run Package

To run a package in the object pane

- Select the package for executing in the object pane.
- Click the  **Execute Package** from the object pane toolbar.  
or
- Right-click and select  **Execute Package** from the popup menu.
- View the returned data on the DBMS Output tab.

To run a package in the Package Designer

- Create a new package/open the existing package.
- Click  **Run**.
- View the returned data on the DBMS Output tab.




## Debug Package

To debug a package h

- Select the package for debugging in the object pane.
- Right-click and select the **Debug Package** from the popup menu.
- Debug the package in the Debugger.

To debug a package in the Package Designer

- Create a new package/open the existing package.
- Click  **Debug**.
- Debug the package in the Debugger.

## Maintain Package

To maintain a package

- Select the package for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Compile
  - Compile Debug

## Achieve Package Information


To achieve a package information

- Select the package in the object pane.
- Right-click the selected package and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Package Designer

**Package Designer** is the basic Navicat tool for working with packages. It allows you to create new package and edit the existing package definition.

- [Editing Package Definition](#)
- Package SQL Preview
- [Viewing Package Result](#)
- [Debugging Package](#)







After saving the package, you can edit the Package Body. Just click  **Design Package Body** to open the Package Body Designer.

## Editing Oracle Package Definition

Edit the package definition under the **Definition** tab.

The **Code Outline** window displays information about the package including function, procedure, parameter, code body, etc. To show the **Code Outline** window, simply choose View -> **Code Outline**.

**Note:** Available only in Full Version.

	Refresh the code outline.
	Show the detail view of the code outline.
	Turn mouse over highlight on or off.
	Expand the selected item.
	Collapse the selected item.
	Toggle sorting by position.

The SQL statements for creating packages are CREATE PACKAGE. In practice, it is best to use a CREATE OR REPLACE statement. The general form of these statements follows.

```
CREATE OR REPLACE
PACKAGE /*PACKAGE NAME*/ AS
    /* TODO enter package declarations (types, exceptions, methods etc) here */
end;
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.

## Oracle Package Body Designer

**Package Body Designer** is the basic Navicat tool for working with package bodies. It allows you to create new package body and edit the existing package body definition.

- [Editing Package Body Definition](#)
- Package Body SQL Preview
- [Viewing Package Result](#)
- [Debugging Package](#)







To edit the Package Specification, click  **Design Package Specification** to open the Package Designer.

## Editing Oracle Package Body Definition

Edit the package body definition under the **Definition** tab.

The **Code Outline** window displays information about the package body including function, procedure, parameter, code body, etc. To show the **Code Outline** window, simply choose View -> **Code Outline**.

**Note:** Available only in Full Version.


	Refresh the code outline.
	Show the detail view of the code outline.
	Turn mouse over highlight on or off.
	Expand the selected item.
	Collapse the selected item.
	Toggle sorting by position.

The SQL statements for creating procedures are CREATE PACKAGE BODY. In practice, it is best to use a CREATE OR REPLACE statement. The general form of these statements follows.

```
CREATE OR REPLACE
PACKAGE BODY /*PACKAGE NAME*/ AS
....
END;
```


**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.


## Viewing Oracle Package Result

To run the package click  **Run** on the toolbar. If the SQL statement is correct, the statement will be executed and, if the statement is supposed to return data, the **Message** and **DBMS Output** tabs open with the message log and data returned by the package. If an error occurs while executing the package, execution stops, the appropriate error message is displayed.

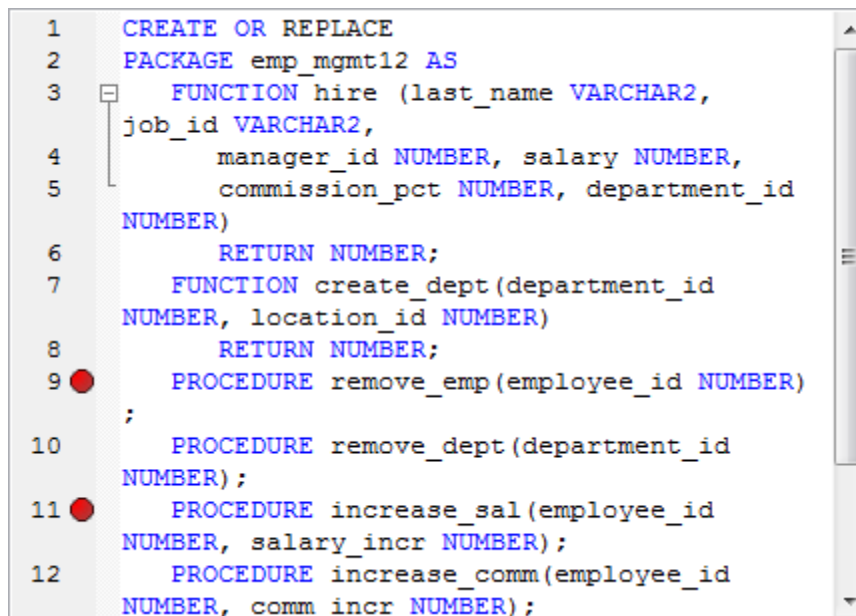
Select the function/procedure and enter the parameter(s) if the function/procedure has input parameter(s).

## Debugging Oracle Package (Available only in Full Version)

To debug the package click  **Debug** on the toolbar to launch the [Oracle Debugger](#).

You can add/remove breakpoints for debugging by clicking  in the grey area beside each statement.

Select the function/procedure and enter the parameter(s) if the function/procedure has input parameter(s).




```
1  CREATE OR REPLACE
2  PACKAGE emp_mgmt12 AS
3  FUNCTION hire (last_name VARCHAR2,
4  job_id VARCHAR2,
5  manager_id NUMBER, salary NUMBER,
6  commission_pct NUMBER, department_id
7  NUMBER)
8  RETURN NUMBER;
9  FUNCTION create_dept(department_id
10 NUMBER, location_id NUMBER)
11 RETURN NUMBER;
12 PROCEDURE remove_emp(employee_id NUMBER)
13 ;
14 PROCEDURE remove_dept(department_id
15 NUMBER);
16 PROCEDURE increase_sal(employee_id
17 NUMBER, salary_incr NUMBER);
18 PROCEDURE increase_comm(employee_id
19 NUMBER, comm incr NUMBER);
```

The screenshot shows a SQL editor window with a package definition. The package is named 'emp\_mgmt12' and contains several functions and procedures. Breakpoints (red circles) are set on the following lines: line 9 (before the 'PROCEDURE remove\_emp' statement), line 11 (before the 'PROCEDURE increase\_sal' statement), and line 12 (before the 'PROCEDURE increase\_comm' statement). The editor has a line number column on the left and a scrollbar on the right.



## Oracle Sequences

Sequence involves creating and initializing a new special single-row table. It is usually used to generate unique identifiers for rows of a table.




Just simply click  -> **Sequence** to open an object pane for **Sequence**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected sequence.

### Create Sequence

To create a new sequence



- Select anywhere on the object pane.
- Click the  **New Sequence** from the object pane toolbar.  
or
- Right-click and select  **New Sequence** from the popup menu.
- Edit sequence properties on the appropriate tabs of the Sequence Designer.

To create a new sequence with modification as one of the existing sequence

- Select the sequence for modifying in the object pane.
- Right-click and select the  **Design Sequence** from the popup menu or simply double-click the sequence.  
or
- Click the  **Design Sequence** from the object pane toolbar.
- Modify sequence properties on the appropriate tabs of the Sequence Designer.
- Click  **Save As**.

### Edit Sequence



To edit the existing sequence(manage its general etc)

- Select the sequence for editing in the object pane.
- Right-click and select the  **Design Sequence** from the popup menu or simply double-click the sequence.  
or
- Click the  **Design Sequence** from the object pane toolbar.
- Edit sequence properties on the appropriate tabs of the Sequence Designer.



## Delete Sequence

To delete a sequence

- Select the sequence for deleting in the object pane.
- Right-click and select the  **Delete Sequence** from the popup menu.  
or
- Click the  **Delete Sequence** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Sequence Information

To achieve a sequence information

- Select the sequence in the object pane.
- Right-click the selected sequence and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Sequence Designer

**Sequence Designer** is the basic Navicat tool for working with sequence. It allows you to create new sequence and edit the existing sequence properties.

- [Editing Sequence General](#)
- Sequence SQL Preview

## Editing Oracle Sequence General

### Increment

To specify which value is added to the current sequence value to create a new value. A positive value will make an ascending sequence, a negative one a descending sequence. The default value is 1.

### Start with

To specify the first sequence number to be generated.

### Minimum

The minimum value a sequence can generate.

### Maximum

The maximum value for the sequence.

### Cache

To specify how many values of the sequence the database preallocates and keeps in memory for faster access. The minimum value for this parameter is 2. (The database caches 20 sequence numbers by default.)

#### ☒ No Cache

This option indicates that values of the sequence are not preallocated.

#### ☒ Cycled


This option allows the sequence continues to generate values after reaching either its maximum or minimum value. After an ascending sequence reaches its maximum value, it generates its minimum value. After a descending sequence reaches its minimum, it generates its maximum value.

#### ☒ Order

This option guarantees that sequence numbers are generated in order of request.



## Oracle Synonyms

Synonym is an alias for any table, view, materialized view, synonym, procedure, function, package, type, Java class schema object, user-defined object type, or another synonym. Because a synonym is simply an alias, it requires no storage other than its definition in the data dictionary.




Just simply click -> **Synonym** to open an object pane for **Synonym**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected synonym.

### Create Synonym

To create a new synonym



- Select anywhere on the object pane.
- Click the  **New Synonym** from the object pane toolbar.  
or
- Right-click and select  **New Synonym** from the popup menu.
- Edit synonym properties on the appropriate tabs of the Synonym Designer.

To create a new synonym with modification as one of the existing synonym

- Select the synonym for modifying in the object pane.
- Right-click and select the  **Design Synonym** from the popup menu or simply double-click the synonym.  
or
- Click the  **Design Synonym** from the object pane toolbar.
- Modify synonym properties on the appropriate tabs of the Synonym Designer.
- Click  **Save As**.

## Edit Synonym

To edit the existing synonym(manage its general etc)



- Select the synonym for editing in the object pane.
- Right-click and select the  **Design Synonym** from the popup menu or simply double-click the synonym.  
or
- Click the  **Design Synonym** from the object pane toolbar.
- Edit synonym properties on the appropriate tabs of the Synonym Designer.

To change the name of the synonym

- Select the synonym for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

## Delete Synonym

To delete a synonym

- Select the synonym for deleting in the object pane.
- Right-click and select the  **Delete Synonym** from the popup menu.  
or
- Click the  **Delete Synonym** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Synonym Information

To achieve a synonym information

- Select the synonym in the object pane.
- Right-click the selected synonym and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Synonym Designer

**Synonym Designer** is the basic Navicat tool for working with synonym. It allows you to create new synonym and edit the existing synonym properties.

- [Editing Synonym General](#)
- Synonym SQL Preview

## Editing Oracle Synonym General

### **Object DataBase Link**

A complete or partial database link to create a synonym for a schema object on a remote database where the object is located.

### **Object Schema**

The schema in which the object resides.

### **Object Type**

The object type.


### **Object**

The object for which the synonym is created.

## Oracle Triggers



Triggers are similar to procedures. A trigger stored in the database can include SQL and PL/SQL or Java statements to run as a unit and can invoke procedures.

See [Triggers](#) for details.




Just simply click  -> **Trigger** to open an object pane for **Trigger**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected trigger.

### Create Trigger

To create a new trigger

- Select anywhere on the object pane.
- Click the  **New Trigger** from the object pane toolbar.  
or
- Right-click and select  **New Trigger** from the popup menu.
- Edit trigger properties on the appropriate tabs of the Trigger Designer.



To create a new trigger with modification as one of the existing trigger

- Select the trigger for modifying in the object pane.
- Right-click and select the  **Design Trigger** from the popup menu or simply double-click the trigger.  
or
- Click the  **Design Trigger** from the object pane toolbar.
- Modify trigger properties on the appropriate tabs of the Trigger Designer.
- Click  **Save As**.



## Edit Trigger

To edit the existing trigger (manage its general, advance, etc)

- Select the trigger for editing in the object pane.
- Right-click and select the  **Design Trigger** from the popup menu or simply double-click the trigger.  
or
- Click the  **Design Trigger** from the object pane toolbar.
- Edit trigger properties on the appropriate tabs of the Trigger Designer.

To change the name of the trigger

- Select the trigger for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.



## Maintain Trigger

To maintain a trigger

- Select the trigger for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Enable
  - Disable
  - Compile
  - Compile for Debug

## Delete Trigger

To delete a trigger

- Select the trigger for deleting in the object pane.
- Right-click and select the  **Delete Trigger** from the popup menu.  
or
- Click the  **Delete Trigger** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Trigger Information

To achieve a trigger information

- Select the trigger in the object pane.
- Right-click the selected trigger and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Trigger Designer

**Trigger Designer** is the basic Navicat tool for working with triggers. It allows you to create new trigger and edit the existing trigger definition.

- [Editing Trigger General](#)
- [Setting Advanced Trigger Properties](#)
- [Editing Trigger Definition](#)
- Trigger SQL Preview

## Editing Oracle Trigger General

### TABLE

To define the trigger on the selected table.

### VIEW

To define the trigger on the selected view.

### SCHEMA

To define the trigger on the selected schema.

### DATABASE

To define the trigger on the entire database.

## Oracle Table Trigger

### Trigger Type

Choose the type of trigger: **TABLE**, VIEW, SCHEMA and DATABASE

#### ☒ Enable

An enabled trigger runs its trigger action if a triggering statement is issued and the trigger restriction (if any) evaluates to true.

### Table Schema

The table schema of the trigger.

### Table name

The table you wish to create the trigger.

#### ☒ Compound

A compound trigger is a single trigger on a table that allows you to specify actions for each of four timing points:

Timing Point	Section
Before the triggering statement executes	BEFORE STATEMENT
After the triggering statement executes	AFTER STATEMENT
Before each row that the triggering statement affects	BEFORE EACH ROW
After each row that the triggering statement affects	AFTER EACH ROW

**Note:** Support from Oracle 11g or later and you can edit the SQL in Trigger Definition.

### Fire

When defining a trigger, you can specify the trigger timing - whether the trigger action is to be run **BEFORE** or **AFTER** the triggering statement.

### For Each

Oracle Database fires a **ROW** trigger once for each row that is affected by the triggering statement and fires a **STATEMENT** trigger only once when the triggering statement is issued if the optional trigger constraint is met.

### When

To specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger.

## On Event

It indicates the kind of statement that activates the trigger.

### ☒ **Insert**

The trigger is activated whenever adding a row to a table or adds an element to a nested table.

### ☒ **Delete**

The trigger is activated whenever removing a row from the table or removes an element from a nested table.

### ☒ **Update**

The trigger is activated whenever changing a value in one of the fields selected in **Update Of Fields**.

## Oracle View Trigger

### Trigger Type

Choose the type of trigger: TABLE, **VIEW**, SCHEMA and DATABASE

#### ☒ **Enable**

An enabled trigger runs its trigger action if a triggering statement is issued and the trigger restriction (if any) evaluates to true.

### Table Schema

The table schema of the trigger.

### Table name

The table you wish to create the trigger.

#### ☒ **Nested Table Field**

To select the nested table field.

#### ☒ **Compound**

To specify the Instead Of Trigger.

**Note:** Support from Oracle 11g or later and you can edit the SQL in Trigger Definition.

### On Event

It indicates the kind of statement that activates the trigger.

#### ☒ **Insert**

The trigger is activated whenever adding a row to a table or adds an element to a nested table.

#### ☒ **Delete**

The trigger is activated whenever removing a row from the table or removes an element from a nested table.

#### ☒ **Update**

The trigger is activated whenever changing a value in a row.

## Oracle Schema Trigger

### Trigger Type

Choose the type of trigger: TABLE, VIEW, **SCHEMA** and DATABASE

### ☒ Enable

An enabled trigger runs its trigger action if a triggering statement is issued and the trigger restriction (if any) evaluates to true.

### Table Schema

The table schema of the trigger.

### Fire

When defining a trigger, you can specify the trigger timing - whether the trigger action is to be run **BEFORE** or **AFTER** the triggering statement.

### When

To specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger.

### Events

The kind of statement that activates the trigger.



## Oracle Database Trigger

### Trigger Type

Choose the type of trigger: TABLE, VIEW, SCHEMA and **DATABASE**

### ☒ Enable

An enabled trigger runs its trigger action if a triggering statement is issued and the trigger restriction (if any) evaluates to true.

### Fire

When defining a trigger, you can specify the trigger timing - whether the trigger action is to be run **BEFORE** or **AFTER** the triggering statement.

### When

To specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger.

### Events

The kind of statement that activates the trigger.

## Setting Advanced Oracle Trigger Properties

The **Advanced** tab is available when the trigger type is TABLE or VIEW.

### Referencing Old

Correlation names of the old nested table.

### Referencing New

Correlation names of the new nested table.

### Referencing Parent

Correlation names of the parent table.

### Follows

To indicate that the trigger should fire after the specified triggers.

Use the **Schema** dropdown list to select the schema name and **Trigger** dropdown list to select the trigger.

To add triggers, just simply press plus button. To remove triggers, select a trigger and press the cross button.


**Note:** Support from Oracle 11g or later.

## Editing Oracle Trigger Definition

The **Definition** tab allows you to edit valid SQL or procedure statements in the trigger definition inside *BEGIN* and *END*.



## Oracle Types

Type is an user-defined datatype that model the structure and behavior of the data in an application. An object type consists of two parts: a specification and a body. The type body always depends on its type specification. A collection type is a named varying array (varray) or a nested table type.

Just simply click  -> **Type** to open an object pane for **Type**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected type.



### Create Object Type (Object Type Specification)

To create a new object type

- Select anywhere on the object pane.
- Click the  **New Object Type** from the object pane toolbar.  
or
- Right-click and select  **New Object Type** from the popup menu.
- Edit object type properties on the appropriate tabs of the Object Type Designer.

### Edit Object Type



To edit the existing object type (manage its definition etc)

- Select the object type for editing in the object pane.
- Right-click and select the  **Design Type** from the popup menu or simply double-click the object type.  
or
- Click the  **Design Type** from the object pane toolbar.
- Edit object type properties on the appropriate tabs of the Object Type Designer.




## Delete Object Type

To delete an object type

- Select the object type for deleting in the object pane.
- Right-click and select the  **Delete Type** from the popup menu.  
or
- Click the  **Delete Type** from the object pane toolbar.
- Confirm deleting in the dialog window.


## Create Type Body (Object Type Body)

To create a new type body

- Select the object type for modifying in the object pane.
- Right-click and select  **New Type Body** from the popup menu.
- Edit type body properties on the appropriate tabs of the Type Body Designer.


## Edit Type Body

To edit the existing type body (manage its definition etc)

- Select the object type for editing in the object pane.
- Right-click and select the  **Design Type Body** from the popup menu.
- Edit type body properties on the appropriate tabs of the Type Body Designer.



## Delete Type Body

To delete a type body




- Select the object type for deleting in the object pane.
- Right-click and select the  **Delete Type Body** from the popup menu.
- Confirm deleting in the dialog window.

## Create Collection Type

To create a new collection type



- Select anywhere on the object pane.
- Click the  **New Collection Type** from the object pane toolbar.  
or
- Right-click and select  **New Collection Type** from the popup menu.
- Edit collection type properties on the appropriate tabs of the Collection Type Designer.

To create a new collection type with modification as one of the existing collection type

- Select the collection for modifying in the object pane.
- Right-click and select the  **Design Type** from the popup menu or simply double-click the collection type.  
or
- Click the  **Design Type** from the object pane toolbar.
- Modify collection type properties on the appropriate tabs of the Collection Type Designer.
- Click  **Save As**.



## Edit Collection Type

To edit the existing collection type(manage its general etc)

- Select the collection type for editing in the object pane.
- Right-click and select the  **Design Type** from the popup menu or simply double-click the collection type.  
or
- Click the  **Design Type** from the object pane toolbar.
- Edit collection type properties on the appropriate tabs of the Collection Type Designer.

## Delete Collection Type

To delete a collection type

- Select the collection type for deleting in the object pane.
- Right-click and select the  **Delete Type** from the popup menu.  
or
- Click the  **Delete Type** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Maintain Type

To maintain a type

- Select the type for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Compile
  - Compile for Debug

## Achieve Type Information


To achieve a type information

- Select the type in the object pane.
- Right-click the selected type and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Object Type Designer

**Object Type Designer** is the basic Navicat tool for working with object types. It allows you to create new object type and edit the existing object type definition.

- [Editing Object Type Definition](#)
- Object Type SQL Preview

After saving the object type, you can edit the Object Type Body. Just click  **Design Object Type Body** to open the Type Body Designer.









## Editing Oracle Object Type Definition

Edit the object type definition under the **Definition** tab.

The **Code Outline** window displays information about the object type including declaration, etc. To show the **Code Outline** window, simply choose View -> **Code Outline**.

**Note:** Available only in Full Version.

	Refresh the code outline.
	Show the detail view of the code outline.
	Turn mouse over highlight on or off.
	Expand the selected item.
	Collapse the selected item.
	Toggle sorting by position.

The SQL statements for creating object types are CREATE TYPE. In practice, it is best to use a CREATE OR REPLACE statement. The general form of these statements follows.

```
CREATE OR REPLACE
TYPE /*TYPE NAME*/ AS OBJECT (
    /* TODO enter type specification (methods, attributes etc) here */
)
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.

## Oracle Type Body Designer

**Type Body Designer** is the basic Navicat tool for working with object type bodies. It allows you to create new type body and edit the existing type body definition.

- [Editing Type Body Definition](#)
- Type Body SQL Preview







To edit the Object Type Specification, click  **Design Object Type Specification** to open the Object Type Designer.

## Editing Oracle Type Body Definition

Edit the type body definition under the **Definition** tab.

The **Code Outline** window displays information about the type body including declaration, etc. To show the **Code Outline** window, simply choose View -> **Code Outline**.

**Note:** Available only in Full Version.

	Refresh the code outline.
	Show the detail view of the code outline.
	Turn mouse over highlight on or off.
	Expand the selected item.
	Collapse the selected item.
	Toggle sorting by position.

The SQL statements for creating type bodies are CREATE TYPE BODY. In practice, it is best to use a CREATE OR REPLACE statement. The general form of these statements follows.

```
CREATE OR REPLACE
TYPE BODY /*TYPE NAME*/ AS
....
END;
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.

## Oracle Collection Type Designer

**Collection Type Designer** is the basic Navicat tool for working with collection types. It allows you to create new collection type and edit the existing collection type definition.

- [Editing Collection Type General](#)
- Collection Type SQL Preview

## Editing Oracle Collection Type General

### **Nested table**

Create a nested table type.

### **VArray**

Create a varray type.

### **Array Size**

Determine the array size of the varray type.

### **Data Type**


Select the Oracle Database built-in datatype or user-defined type of the attribute.

### **Data Type Parameter**

Determine the corresponding data type parameters.



## Oracle XML Schemas

XML Schema is a schema definition language written in XML. It can be used to describe the structure and various other semantics of conforming instance documents.

Just simply click  -> **XML Schema** to open an object pane for **XML Schema**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected XML Schema.

### Create XML Schema

To create a new XML Schema

- Select anywhere on the object pane.
- Click the  **New XML Schema** from the object pane toolbar.  
or
- Right-click and select  **New XML Schema** from the popup menu.
- Edit XML Schema properties on the appropriate tabs of the XML Schema Designer.



### Maintain XML Schema

To maintain an XML Schema

- Select the XML Schema for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Compile
  - Purge

### Delete XML Schema

To delete an XML Schema

- Select the XML Schema for deleting in the object pane.
- Right-click and select the  **Delete XML Schema** from the popup menu.  
or
- Click the  **Delete XML Schema** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve XML Schema Information

To achieve an XML Schema information

- Select the XML Schema in the object pane.
- Right-click the selected XML Schema and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle XML Schema Designer

**XML Schema Designer** is the basic Navicat tool for working with XML Schemas. It allows you to create new XML Schema.

- [Editing XML Schema Doc](#)
- [Setting Advanced XML Schema Properties](#)
- XML Schema SQL Preview



## Editing Oracle XML Schema Doc

Enter a valid XML schema document under the **Schema Doc** tab. The general form follows.

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
</xsd:schema>
```

## Setting Advanced Oracle XML Schema Properties

### ☒ **Local**

Check this to register as local schemas.

### ☒ **Force on schema registration**

Check this to ignore errors generated during schema evolution.

## Generate

### ☒ **Object Types**

Check this to enable the schema compiler to generate object types.

### ☒ **Java Beans**

Check this to enable the schema compiler to generate Java beans.

### ☒ **Default Tables**

Check this to enable the schema compiler to generate default tables.

## Options

### ☒ **REGISTER\_NODOCID**

Check this to prevent the creation of this column if the user wishes to optimize on storage.

### ☒ **REGISTER\_BINARYXML**

Check this to register the schema for Binary XML.

### ☒ **REGISTER\_NT\_AS\_IOT**

Check this to store nested tables created during schema registration as index organized tables.

### ☒ **REGISTER\_AUTO\_OOL**

Check this to automatically move large types out of line.

## Enable Hierarchy

### **ENABLE\_HIERARCHY\_NONE**

Enable hierarchy will not be called on any tables created while registering that schema.

## **ENABLE\_HIERARCHY\_CONTENTS**


Enable hierarchy will be called for all tables created during schema registration with hierarchy\_type as DBMS\_XDBZ.ENABLE\_CONTENTS.

## **ENABLE\_HIERARCHY\_RESMETADATA**

Enable hierarchy will be called on all tables created during schema registration with hierarchy\_type as DBMS\_XDBZ.ENABLE\_RESMETADATA.



## Oracle Recycle Bin

Recycle bin is actually a data dictionary table containing information about dropped objects. Dropped tables and any associated objects such as indexes, constraints, nested tables, and the likes are not removed and still occupy space. They continue to count against user space quotas, until specifically purged from the recycle bin or the unlikely situation where they must be purged by the database because of tablespace space constraints.

Just simply click  -> **Recycle Bin** to open an object pane for **Recycle Bin**. A right-click displays the popup menu or using the object pane toolbar, allowing you to flashback tables or purge the deleted objects.



### Restore tables

To restore table from recycle bin

- Choose a table in recycle bin.
- Click the  **Flashback Table** from the object pane toolbar.  
or
- Right-click and select  **Flashback Table** from the popup menu.

### Purge Objects

To remove an object in the recycle bin

- Select an object for purging in the object pane.
- Right-click and select the  **Purge Object** from the popup menu.  
or
- Click the  **Purge Object** from the object pane toolbar.
- Confirm deleting in the dialog window.

To remove all objects in the recycle bin

- Right-click and select the **Purge Recycle Bin** from the popup menu.
- Confirm deleting in the dialog window.

To remove all objects in the recycle bin of every user

- Log in a user has the **SYSDBA** privilege.
- Right-click and select the **Purge DBA Recycle Bin** from the popup menu.
- Confirm deleting in the dialog window.


## **Achieve Recycle Bin Object Information**

To achieve an object information in recycle bin

- Select the object in the object pane.
- Right-click the selected object and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.



## Oracle Directories

A directory object specifies an alias for a directory on the server file system where external binary file LOBs (BFILEs) and external table data are located. All directories are created in a single namespace and are not owned by an individual schema.




Just simply click -> **Directory** to open an object pane for **Directory**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected directory.

### Create Directory

To create a new directory



- Select anywhere on the object pane.
- Click the  **New Directory** from the object pane toolbar.  
or
- Right-click and select  **New Directory** from the popup menu.
- Edit directory properties on the appropriate tabs of the Directory Designer.

To create a new directory with modification as one of the existing directory

- Select the directory for modifying in the object pane.
- Right-click and select the  **Design Directory** from the popup menu or simply double-click the directory.  
or
- Click the  **Design Directory** from the object pane toolbar.
- Modify directory properties on the appropriate tabs of the Directory Designer.
- Click  **Save As**.



### Edit Directory

To edit the existing directory(manage its general etc)

- Select the directory for editing in the object pane.
- Right-click and select the  **Design Directory** from the popup menu or simply double-click the directory.  
or
- Click the  **Design Directory** from the object pane toolbar.
- Edit directory properties on the appropriate tabs of the Directory Designer.

## Delete Directory

To delete a directory

- Select the directory for deleting in the object pane.
- Right-click and select the  **Delete Directory** from the popup menu.  
or
- Click the  **Delete Directory** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Directory Information

To achieve a directory information

- Select the directory in the object pane.
- Right-click the selected directory and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Directory Designer

**Directory Designer** is the basic Navicat tool for working with directories. It allows you to create new directory and edit the existing directory definition.

- [Editing Directory General](#)
- Directory SQL Preview



## Editing Oracle Directory General

Edit the directory general properties under the **General** tab.


### Directory Path

Specify the full path name of the operating system directory of the server where the files are located. The path name is case sensitive





## Oracle Tablespaces

Tablespaces are the allocation of space in the database that can contain schema objects.




Just simply click  -> **Tablespace** to open an object pane for **Tablespace**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected tablespace.

### Create Tablespace

To create a new tablespace



- Select anywhere on the object pane.
- Click the  **New Tablespace** from the object pane toolbar.  
or
- Right-click and select  **New Tablespace** from the popup menu.
- Edit tablespace properties on the appropriate tabs of the Tablespace Designer.

To create a new tablespace with modification as one of the existing tablespace

- Select the tablespace for modifying in the object pane.
- Right-click and select the  **Design Tablespace** from the popup menu or simply double-click the tablespace.  
or
- Click the  **Design Tablespace** from the object pane toolbar.
- Modify tablespace properties on the appropriate tabs of the Tablespace Designer.
- Click  **Save As**.

### Edit Tablespace

To edit the existing tablespace (manage its properties etc)

- Select the tablespace for editing in the object pane.
- Right-click and select the  **Design Tablespace** from the popup menu or simply double-click the tablespace.  
or
- Click the  **Design Tablespace** from the object pane toolbar.
- Edit tablespace properties on the appropriate tabs of the Tablespace Designer.

To change the name of the tablespace

- Select the tablespace for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.



## Maintain Tablespace

To maintain a tablespace

- Select the tablespace for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - [Read Only]
  - Read Write
  - Online
  - Offline
    - Normal
    - Temporary
    - Immediate
  - Coalesce
  - Shrink Space

## Delete Tablespace

To delete a tablespace

- Select the tablespace for deleting in the object pane.
- Right-click and select the  **Delete Tablespace** from the popup menu.  
or
- Click the  **Delete Tablespace** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Tablespace Information

To achieve a tablespace information

- Select the tablespace in the object pane.
- Right-click the selected tablespace and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Oracle Tablespace Designer

**Tablespace Designer** is the basic Navicat tool for working with tablespaces. It allows you to create new tablespace and edit the existing tablespace properties.

- [Editing Tablespace General](#)
- [Editing Tablespace Storage](#)
- [Setting Advanced Tablespace Properties](#)
- Tablespace SQL Preview

## Editing Oracle Tablespace General

### Tablespace Type

#### PERMANENT

A permanent tablespace contains persistent schema objects. Objects in permanent tablespaces are stored in datafiles.

#### TEMPORARY

A temporary tablespace contains schema objects only for the duration of a session. Objects in temporary tablespaces are stored in tempfiles.

#### UNDO

An undo tablespace is a type of permanent tablespace used by Oracle Database to manage undo data if you are running your database in automatic undo management mode.

### Data File

Use the **Name** and **Size** edit box to set the datafile / tempfile name and size.

The **Unit** drop-down list defines the unit of the size of the datafile / tempfile. Specify the maximum disk space allowed for automatic extension of the datafile. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.

#### ☒ Reuse

To allow Oracle to reuse an existing file.

### Path

Specify the path of the datafile / tempfile.

### Auto Extend

To **ON** (enable) or **OFF** (disable) the automatic extension of a new or existing datafile or tempfile.

## **Next Size**

Specify the size in bytes of the next increment of disk space to be allocated automatically when more extents are required. The default is the size of one data block. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.

## ☒ **Unlimited Max**

Unlimited disk space that Oracle can allocate to the datafile or tempfile.

## **Max Size**

Specify the maximum disk space allowed for automatic extension of the datafile. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.

## Editing Oracle Tablespace Storage

### File Type

#### **BIGFILE**

A bigfile tablespace contains only one datafile or tempfile, which can contain up to approximately 4 billion ( $2^{32}$ ) blocks. The maximum size of the single datafile or tempfile is 128 terabytes (TB) for a tablespace with 32K blocks and 32TB for a tablespace with 8K blocks.

#### **SMALLFILE**

A smallfile tablespace is a traditional Oracle tablespace, which can contain 1022 datafiles or tempfiles, each of which can contain up to approximately 4 million ( $2^{22}$ ) blocks.

### Min Extent Size

The minimum size of an extent in the tablespace. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.

### Block Size

The block size for the tablespace.

### [Default Storage Options](#)

Set the default storage characteristics for objects created in the tablespace.

### Table Compression

Use the dropdown list to select the type of compressing data segments to reduce disk use.

#### ☒ **Manual Segment Management**

To manage the free space of segments in the tablespace using free lists.

### Extent Management

To specify how the extents of the tablespace will be managed.

#### **Extent Management**

##### **DICTIONARY**

Extent management by the data dictionary.

## **LOCAL**

Extent management by the bitmaps.

### **Local Extent**

## **AUTOALLOCATE**

The tablespace is system managed.

## **UNIFORM**

The tablespace is managed with uniform extents of size.

### **Uniform Size**

The size of uniform extent. The default size is 1 megabyte. Use the dropdown list K, M, G, T, P or E to specify the size in kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes. If you do not specify any of the abbreviations, then the size is interpreted as bytes.



## Setting Advanced Oracle Tablespace Properties

### Logging

#### **LOGGING**

Log all objects within the tablespace in the redo log file.

#### **NOLOGGING**

No operations are logged.

#### ☒ **Force Logging**

Oracle Database will log all changes to all objects in the tablespace except changes to temporary segments, overriding any NOLOGGING setting for individual objects.

#### ☒ **Offline**

The tablespace is unavailable immediately (offline) after creation.

#### ☒ **Retention Guarantee**

Oracle Database should preserve unexpired undo data in all undo segments of tablespace even if doing so forces the failure of ongoing operations that need undo space in those segments.

### Tablespace Group

To determine whether tablespace is a member of a tablespace group.

### Flashback

#### **ON**

Oracle Database will save Flashback log data for this tablespace and the tablespace can participate in a FLASHBACK DATABASE operation.

#### **OFF**

Oracle Database will not save any Flashback log data for this tablespace.

### Encryption

#### ☒ **Use Encryption**

Enable the encryption properties of the tablespace.

#### **Algorithm**

To select the encryption algorithm.

## Oracle Public Database Links

Public database Link is a database link created by a *DBA* on a local database that is accessible to all users on that database.

See [Database Link](#) for details.

## Oracle Public Synonyms

Public synonym is a synonym owned by the special user group named *PUBLIC* and every user in a database can access it.

See [Synonyms](#) for details.

## PostgreSQL Database Object Management

The following list contains the most common PostgreSQL database objects supported by Navicat.

- [Schemas](#)
- [Tables](#)
- [Views](#)
- [Functions](#)
- [Aggregates](#)
- [Conversions](#)
- [Domains](#)
- [Trigger Functions](#)
- [Operators](#)
- [Operator Class](#)
- [Sequences](#)
- [Types](#)
- [Tablespaces](#)
- [Casts](#)
- [Languages](#)



## PostgreSQL Schemas

A schema is essentially a namespace: it contains named objects (tables, data types, functions, and operators) whose names may duplicate those of other objects existing in other schemas.

The schema name must be distinct from any existing schema name in the current database.


### Create Schema

To create a new schema

- Right-click the database in the navigation pane and choose  **New Schema....**  
or
- Right-click any existing schema and choose  **New Schema....**
- Edit schema properties on the appropriate tabs of the Schema Designer.


### Edit Schema

To edit the existing schema(manage its general etc)

- Right-click the schema in the navigation pane and choose  **Schema Properties....**
- Edit schema properties on the appropriate tabs of the Schema Designer.


### Open Schema

To open a schema which shows in the navigation pane

- Double-click the schema to open in the navigation pane.  
or
- Right-click the schema and choose  **Open Schema.**


### Close Schema

To close a schema

- Right-click the schema in the navigation pane and choose  **Close Schema.**

## Delete Schema

To delete a schema

- Right-click the schema in the navigation pane and choose  **Delete Schema**.
- Confirm deleting in the dialog window.

## PostgreSQL Schema Designer

**Schema Designer** is the basic Navicat tool for working with schema. It allows you to create new schema and edit the existing schema properties.

- [Editing Schema General](#)
- Editing Schema Comment

## Editing PostgreSQL Schema General

### Schema Name

The name of a schema to be created. The name cannot begin with pg\_, as such names are reserved for system schemas.


### Owner

The name of the user who will own the schema. If omitted, defaults to the user executing the command.





## PostgreSQL Tables


Relational databases use tables to store data. All operations on data are done on the tables themselves or produce another tables as the result. A table is a set of rows and columns, and their intersections are fields. From a general perspective, columns within a table describe the name and type of data that will be found by row for that column's fields. Rows within a table represent records composed of fields that are described from left to right by their corresponding column's name and type. Each field in a row is implicitly correlated with each other field in that row.

Just simply click  to open an object pane for **Table**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit, open and delete the selected table.

### Create Table

To create a new table

- Select anywhere on the object pane.
- Click the  **New Table** from the object pane toolbar.  
or
- Right-click and select  **New Table** from the popup menu.
- Edit table properties and fields on the appropriate tabs of the Table Designer.

**Hint:** To create new table you can also right-click the Tables node of the navigation pane and select the  **New Table** from the popup menu.

To create a new table with the same properties as one of the existing tables has (using popup menu)

**Apply to:** current database {same connection}

- Select the table(s) for copying in the navigation pane/object pane.
- Right-click and select the **Duplicate Table** from the popup menu.
- The newly created table(s) will be named as "tablename\_**copy**".

To create a new table with the same properties as one of the existing tables has (using drag and drop method)

**Apply to:** current database {same connection}




- Select the table(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen table(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created table(s) will be named as "tablename\_**copy**"

**Apply to:** different database {same connection}

different database {different connection (same or cross server type)} (Data Transfer tool will be activated)

- Select the table(s) for copying in the object pane.
- Drag and drop the chosen table(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel

To create a new table with modification as one of the existing tables

- Select the table for modifying in the navigation pane/object pane.
- Right-click and select the  **Design Table** from the popup menu.  
or
- Click the  **Design Table** from the object pane toolbar.
- Modify table properties and fields on the appropriate tabs of the Table Designer.
- Click  **Save As**.

## Create Table Shortcut



To create a table shortcut

- Select the table for editing in the navigation pane/object pane.
- Right-click and select **Create Open Table Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.

**Note:** This option is used to provide a convenient way for you to open your table for entering data directly (Grid View/Form View) without activating the main Navicat.

## Edit Table

To edit the existing table (manage its fields, indexes, foreign keys and triggers etc)



- Select the table for editing in the navigation pane/object pane.
- Right-click and select the  **Design Table** from the popup menu.  
or
- Click the  **Design Table** from the object pane toolbar.
- Edit table properties and fields on the appropriate tabs of the Table Designer.


To change the name of the table

- Select the table for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.


## Open Table (manage table data)

To open a table

- Select the table for opening in the navigation pane/object pane.
- Right-click and select the  **Open Table** from the popup menu or simply double-click the table.  
or
- Click the  **Open Table** from the object pane toolbar.

**Note:** This option is only applied if you do wish Navicat loads all your images while opening the table. To open the graphical table with faster performance, use  **Open Table (Quick)** below.

To open a table with graphical fields

- Select the table for opening in the navigation pane/object pane.
- Right-click and select the  **Open Table (Quick)** from the popup menu.

**Note:** Faster performance for opening the graphical table, as BLOB fields (images) will not be loaded until you click on the cell.

## Empty Table

To empty a table

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Empty Table** from the popup menu.

**Note:** This option is only applied when you wish to clear all the existing records without resetting the auto-increment value. To reset the auto-increment value while emptying your table, use **Truncate Table** below.



## Truncate Table

To truncate a table

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Truncate Table** from the popup menu.

## Delete Table

To delete a table

- Select the table for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Table** from the popup menu.  
or
- Click the  **Delete Table** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Table Information

To achieve a table information

- Select the table in the object pane.
- Right-click the selected table and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## PostgreSQL Table Designer

**Table Designer** is the basic Navicat tool for working with tables. It allows you to create, edit and drop table's fields, indexes, foreign keys, and much more.



- [Managing Table Fields](#)
- [Managing Table Indexes](#)
- [Managing Table Foreign Keys](#)
- [Managing Table Uniques](#)
- [Managing Table Checks](#)
- [Managing Table Excludes](#)
- [Managing Table Rules](#)
- [Managing Table Triggers](#)
- [Managing Table Options](#)
- Managing Table Comment
- Table SQL Preview

## PostgreSQL Table Fields

Table fields are managed on the **Fields** tab of the Table Designer. Just simply click a field for editing. A right-click displays the popup menu or using the field toolbar, allowing you to create new and drop the selected field.

### Add Field

To add a field to the table

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click and select the  **Add Field** from the popup menu or click the  **Add Field** from the toolbar.
- Edit field properties.

To add a new field with modification as one of the existing fields

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Select field.
- Right-click and select the **Duplicate Field** from the popup menu.
- Edit field properties.



### Edit Field

To edit the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Simply click on the field to edit.


## Delete Field

To delete the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click and select the  **Delete Field** from the popup menu or click the  **Delete Field** from the toolbar.
- Confirm deleting in the dialog window.



## Setting PostgreSQL Table Field Properties

Name	Type	Length	Decimals	Allow Null	
▶ CustNo	float8	53	0	<input type="checkbox"/>	 1
Company	varchar	30	0	<input checked="" type="checkbox"/>	
Addr1	text	0	0	<input checked="" type="checkbox"/>	
Addr2	text	0	0	<input checked="" type="checkbox"/>	

### Name

The Name is a descriptive identifier for a field that can be up to 63 bytes long. The names should be descriptive enough that anyone can easily identify them when viewing or editing records. For example, LastName, FirstName, StreetAddress, or HomePhone.

Use the **Name** edit box to set the field name. Note that the name of the field must be unique among all the field names in the table.

### Type

After you name a field, you choose a data type for the data to be contained in the field. When you choose a field's data type, you are deciding:

- What kind of values to allow in the field. You cannot store text in field with the **Numeric** data type.
- How much storage space PostgreSQL is to set aside for the data in that field.
- What types of operations can be performed on the values in that field.

The **Type** dropdown list defines the type of the field data.

The following table shows the built-in general-purpose data types for PostgreSQL 8.3. Most of the alternative names listed in the "Aliases" column are the names used internally by PostgreSQL for historical reasons.

**Note:** Some built-in general-purpose data types are not applicable for PostgreSQL 8.2 or earlier versions.

Name	Aliases	Description
bigint	int8	signed eight-byte integer
bigserial	serial8	autoincrementing eight-byte integer
bit [ (n) ]		fixed-length bit string

bit varying [ (n) ]	varbit	variable-length bit string
boolean	bool	logical Boolean (true/false)
box		rectangular box in the plane
bytea		binary data ("byte array")
character varying [ (n) ]	varchar [ (n) ]	variable-length character string
character [ (n) ]	char [ (n) ]	fixed-length character string
cidr		IPv4 or IPv6 network address
circle		circle in the plane
date		calendar date (year, month, day)
double precision	float8	double precision floating-point number
inet		IPv4 or IPv6 host address
integer	int, int4	signed four-byte integer
interval [ (p) ]		time span
line		infinite line in the plane
lseg		line segment in the plane
macaddr		MAC address
money		currency amount
numeric [ (p, s) ]	decimal [ (p, s) ]	exact numeric of selectable precision
path		geometric path in the plane
point		geometric point in the plane
polygon		closed geometric path in the plane
real	float4	single precision floating-point number
smallint	int2	signed two-byte integer
serial	serial4	autoincrementing four-byte integer
text		variable-length character string
time [ (p) ] [ without time zone ]		time of day
time [ (p) ] with time zone	timetz	time of day, including time zone
timestamp [ (p) ] [ without time zone ]		date and time
timestamp [ (p) ] with time zone	timestampz	date and time, including time zone

tsquery		text search query
tsvector		text search document
txid_snapshot		user-level transaction ID snapshot
uuid		universally unique identifier
xml		XML data

## Length and Decimals

Use the **Length** edit box to define the length of the field and use **Decimals** edit box to define the number of digits after the decimal point (the scale) for Floating Point data type.

**Note:** Be careful when shortening the field length as losing data might be caused.

### **Allow Null**

Allow the NULL values for the field.

### **Primary Key**

A Primary Key is a single field or combination of fields that uniquely defines a record. None of the fields that are part of the primary key can contain a null value.

### **Primary Key Name**

Right-click and select **Primary Key Name** from the popup menu to enter the primary key constraint name.

### **Fill Factor**

Right-click and select **Fill Factor** from the popup menu to enter the storage parameter. The fillfactor for a table is a percentage between 10 and 100.

## Setting Other PostgreSQL Table Field Properties

To set the default value for the field use the **Default** edit box.

To set any optional text describing the current field use the **Comment** edit box.

To set the dimensions of array specifiers use the **Dimensions** edit box.

For **Domain** and **Type** data types:

### **Object Schema**

Set the object schema for the field.

### **Object Type**

Set the object type for the field.

## PostgreSQL Table Indexes



Indexes are primarily used to enhance database performance (though inappropriate use can result in slower performance).

An index field can be an expression computed from the values of one or more columns of the table row. This feature can be used to obtain fast access to data based on some transformation of the basic data.

Table indexes are managed on the **Indexes** tab of the Table Designer. Just simply click/double-click an index field for editing. A right-click displays the popup menu or using the index toolbar, allowing you to create new, edit and delete the selected index field.

### Add Index

To add a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Right-click and select the  **Add Index** from the popup menu or click the  **Add Index** from the toolbar.
- Edit index properties.



### Edit Index

To edit a table index


- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Just simply click/double-click on the index to edit.

### Delete Index


To delete a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Right-click on the index to delete and select the  **Delete Index** from the popup menu or click the  **Delete Index** from the toolbar.
- Confirm deleting in the dialog window.

## Setting PostgreSQL Table Index Properties

Name	Fields	Index method	Unique	Clustered
▶ cust_index	CustNo 	B-Tree	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Use the **Name** edit box to set the index name. No schema name can be included here; the index is always created in the same schema as its parent table.

To include field(s) in the index, just simply double-click the **Fields** field or click  to open the editor for editing.

**Note:** Some of field types do not allow indexing by several fields.

The **Index method** dropdown list defines the type of the table index. PostgreSQL provides the index methods B-tree, R-tree, hash, and GiST. The B-tree index method is an implementation of Lehman-Yao high-concurrency B-trees. The R-tree index method implements standard R-trees using Guttman's quadratic split algorithm. The hash index method is an implementation of Litwin's linear hashing. Users can also define their own index methods, but that is fairly complicated.

### **Unique**

Makes index unique, causes the system to check for duplicate values in the table when the index is created (if data already exist) and each time data is added.

### **Clustered**

*CLUSTER* instructs PostgreSQL to cluster the table specified by tablename based on the index specified by indexname. The index must already have been defined on tablename.

When a table is clustered, PostgreSQL remembers on which index it was clustered. The form *CLUSTER* tablename reclusters the table on the same index that it was clustered before.

### **Tablespace**

The tablespace in which to create the index.

### **Constraints**

If you wish to create partial index, enter constraint condition in this edit box. A partial index is an index that contains entries for only a portion of a table, usually a portion that is more useful for indexing than the rest of the table.

The **Comment** edit box defines the comment for the index..



## PostgreSQL Table Foreign Keys

A foreign key specifies that the values in a column (or a group of columns) must match the values appearing in some row of another table. We say this maintains the referential integrity between two related tables.

Foreign Keys are managed on the **Foreign Keys** tab of the Table Designer. Just simply click/double-click a foreign key field for editing. A right-click displays the popup menu or using the foreign key toolbar, allowing you to create new, edit and delete the selected foreign key field.

### Add Foreign Key

To add a foreign key

- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Right-click and select the  **Add Foreign Key** from the popup menu or click the  **Add Foreign Key** from the toolbar.
- Edit foreign key properties.



### Edit Foreign Key

To edit a foreign key


- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Just simply click/double-click on the foreign key to edit.

### Delete Foreign Key

To delete a foreign key


- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Right-click on the foreign key to delete and select the  **Delete Foreign Key** from the popup menu or click the  **Delete Foreign Key** from the toolbar.
- Confirm deleting in the dialog window.

## Setting PostgreSQL Table Foreign Key Properties

Name	Fields	Reference Schema	Reference Table	Reference Fields	On Delete	On Update
▶ orders_cust_fk	CustNo  	report	customer	CustNo	NO ACTION	NO ACTION

Use the **Name** edit box to enter a name for the new key and then select a table field to include in the key from the **Fields** group.

Use the **Reference Schema** and **Reference Table** dropdown lists to select a foreign schema and table respectively.

To include field(s) to the key, just simply double-click the **Fields/Reference Fields** field or click  to open the editor(s) for editing.

The **On Delete** and **On Update** dropdown list define the type of the actions to be taken.

### Restrict

Produce an error indicating that the deletion or update would create a foreign key constraint violation. This is the same as NO ACTION except that the check is not deferrable.

### No Action

Produce an error indicating that the deletion or update would create a foreign key constraint violation. If the constraint is deferred, this error will be produced at constraint check time if there still exist any referencing rows. This is the default action.

### Cascade

Delete any rows referencing the deleted row, or update the value of the referencing column to the new value of the referenced column, respectively.

### Set Null

Set the referencing column(s) to null.

### Set Default

Set the referencing column(s) to their default values.





## PostgreSQL Table Uniques

Unique constraints ensure that the data contained in a column or a group of columns is unique with respect to all the rows in the table.

Uniques are managed on the **Uniques** tab of the Table Designer. Just simply click/double-click an unique field for editing. Using the unique toolbar, allowing you to create new, edit and delete the selected unique field.

### Add Unique

To add an unique

- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Right-click and select the  **Add Unique** from the popup menu or click the  **Add Unique** from the toolbar.
- Edit unique properties.



### Edit Unique

To edit an unique


- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Just simply click on the unique to edit.

### Delete Unique

To delete an unique


- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Right-click on the unique to delete and select the  **Delete Unique** from the popup menu or click the  **Delete Unique** from the toolbar.
- Confirm deleting in the dialog window.

## Setting PostgreSQL Table Unique Properties

Name	Fields
▶ order_cust_index	OrderNo, CustNo 

Use the **Name** edit box to set the unique name.

### Fields

To set field(s) as unique, just simply double-click the **Fields** field or click  to open the editor(s) for editing.

Select the field(s) from the list. To remove the fields from the unique, uncheck them in the same way.

### Tablespace

Allows setting a tablespace different from the default tablespace.

The **Comment** edit box defines the comment for the unique.

### Fill Factor

The fillfactor for a unique is a percentage between 10 and 100. 100 (complete packing) is the default.

**Note:** Support from PostgreSQL 8.2 or later.



## PostgreSQL Table Checks

A check constraint is the most generic constraint type. It allows you to specify that the value in a certain column must satisfy a Boolean (truth-value) expression.

Checks are managed on the **Checks** tab of the Table Designer. Just simply click/double-click a check field for editing. Using the check toolbar, allowing you to create new, edit and delete the selected check field.

### Add Check

To add a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Right-click and select the  **Add Check** from the popup menu or click the  **Add Check** from the toolbar.
- Edit check properties.



### Edit Check

To edit a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Just simply click on the check to edit.

### Delete Check

To delete a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Right-click on the check to delete and select the  **Delete Check** from the popup menu or click the  **Delete Check** from the toolbar.
- Confirm deleting in the dialog window.

## Setting PostgreSQL Table Check Properties

Use the **Name** edit box to set the check name.

### Check

Set the condition for checking, e.g. "field\_name1 > 0 AND field\_name2 > field\_name1" in the **Check** edit box. A check constraint specified as a column constraint should reference that column's value only, while an expression appearing in a table constraint may reference multiple columns.

### Definition

allows you to enter the definition for the check.

### Comment

allows you to enter the comment for the check.

## PostgreSQL Table Excludes



An exclude constraint guarantees that if any two rows are compared on the specified column(s) or expression(s) using the specified operator(s), not all of these comparisons will return TRUE.

Excludes are managed on the **Excludes** tab of the Table Designer. Just simply click/double-click an exclude field for editing. Using the exclude toolbar, allowing you to create new, edit and delete the selected exclude field.

**Note:** Exclude is supported from PostgreSQL 9.0 or later.

### Add Exclude

To add an exclude

- Open the table in the Table Designer.
- Open the **Excludes** tab.
- Right-click and select the  **Add Exclude** from the popup menu or click the  **Add Exclude** from the toolbar.
- Edit exclude properties.



### Edit Exclude

To edit an exclude

- Open the table in the Table Designer.
- Open the **Excludes** tab.
- Just simply click on the exclude to edit.

### Delete Exclude

To delete an exclude

- Open the table in the Table Designer.
- Open the **Excludes** tab.
- Right-click on the exclude to delete and select the  **Delete Exclude** from the popup menu or click the  **Delete Exclude** from the toolbar.
- Confirm deleting in the dialog window.

## Setting PostgreSQL Table Exclude Properties

Use the **Name** edit box to set the exclude name.

### Index method

The name of the index access method to be used.

### Element

Choose the element(s) to be excluded and specify the operator(s).

### Tablespace

The tablespace in which to create the index.

### Fill Factor

The fillfactor for an index is a percentage that determines how full the index method will try to pack index pages.

### Predicate

Specify an exclusion constraint on a subset of the table.

The **Comment** edit box defines the comment for the exclude.

## PostgreSQL Table Rules



The PostgreSQL rule system allows one to define an alternate action to be performed on insertions, updates, or deletions in database tables. Roughly speaking, a rule causes additional commands to be executed when a given command on a given table is executed.

**Note:** You must be the owner of a table to create or change rules for it.

Rules are managed on the **Rules** tab of the Table Designer. Just simply click/double-click a rule field for editing. Using the rule toolbar, allowing you to create new, edit and delete the selected rule field.

### Add Rule

To add a rule

- Open the table in the Table Designer.
- Open the **Rules** tab.
- Right-click and select the  **Add Rule** from the popup menu or click the  **Add Rule** from the toolbar.
- Edit rule properties.



### Edit Rule

To edit a rule

- Open the table in the Table Designer.
- Open the **Rules** tab.
- Just simply click on the rule to edit.

### Delete Rule

To delete a rule

- Open the table in the Table Designer.
- Open the **Rules** tab.
- Right-click on the rule to delete and select the  **Delete Rule** from the popup menu or click the  **Delete Rule** from the toolbar.
- Confirm deleting in the dialog window.

## Setting PostgreSQL Table Rule Properties

Use the **Name** edit box to set the rule name. This must be distinct from the name of any other rule for the same table. Multiple rules on the same table and same event type are applied in alphabetical name order.

### Event

The event is one of *SELECT*, *INSERT*, *UPDATE*, or *DELETE*.

#### ☒ Do instead

This indicates that the commands should be executed instead of the original command. Otherwise, the commands should be executed in addition to the original command.

### Condition

Any SQL conditional expression (returning boolean). The condition expression may not refer to any tables except NEW and OLD, and may not contain aggregate functions.

### Definition

The command or commands that make up the rule action. Valid commands are *SELECT*, *INSERT*, *UPDATE*, *DELETE*, or *NOTIFY*.

Within condition and command, the special table names NEW and OLD may be used to refer to values in the referenced table. NEW is valid in ON INSERT and ON UPDATE rules to refer to the new row being inserted or updated. OLD is valid in ON UPDATE and ON DELETE rules to refer to the existing row being updated or deleted

The **Comment** edit box defines the comment for the rule.



## PostgreSQL Table Triggers



A trigger is a specification that the database should automatically execute a particular function whenever a certain type of operation is performed. Triggers can be defined to execute either before or after any INSERT, UPDATE, or DELETE operation, either once per modified row, or once per SQL statement.

Triggers are managed on the **Triggers** tab of the Table Designer. Just simply click a trigger field for editing. A right-click displays the popup menu or using the trigger toolbar, allowing you to create new, edit and delete the selected trigger field.

**Note:** To create a trigger on a table, the user must have the TRIGGER privilege on the table.

### Add Trigger

To add a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Right-click and select the  **Add Trigger** from the popup menu or click the  **Add Trigger** from the toolbar.
- Edit trigger properties.



### Edit Trigger

To edit a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Just simply click on the trigger to edit.

### Delete Trigger

To delete a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Right-click on the trigger to delete and select the  **Delete Trigger** from the popup menu or click the  **Delete Trigger** from the toolbar.
- Confirm deleting in the dialog window.

## Setting PostgreSQL Table Trigger Properties

Use the **Name** edit box to set the trigger name. This must be distinct from the name of any other trigger for the same table.

### Row trigger

This specifies whether the trigger procedure should be fired once for every row affected by the trigger event, or just once per SQL statement. If unchecked, *FOR EACH STATEMENT* is the default.

Use the **Fires** dropdown list to define the trigger action time. It can be **Before** or **After** to indicate that the trigger activates before or after the statement that activated it.

### Insert

The trigger is activated whenever a new row is inserted into the table.

### Update

The trigger is activated whenever a row is modified.

### Delete

The trigger is activated whenever a row is deleted from the table.

### Update Of Fields

Specify a list of columns. The trigger will only fire if at least one of the listed columns is mentioned as a target of the UPDATE command.

**Note:** Support from PostgreSQL 9.1 or later.

### When Clause

Specify a Boolean WHEN condition, which will be tested to see whether the trigger should be fired.

**Note:** Support from PostgreSQL 9.0 or later.

### Trigger Function Schema and Trigger Function

A user-supplied function that is declared as taking no arguments and returning type trigger, which is executed when the trigger fires.

## Arguments

An optional comma-separated list of arguments to be provided to the function when the trigger is executed. The arguments are literal string constants. Simple names and numeric constants may be written here, too, but they will all be converted to strings. Please check the description of the implementation language of the trigger function about how the trigger arguments are accessible within the function; it may be different from normal function arguments.

The **Comment** edit box defines the comment for the trigger.

## PostgreSQL Table Options


The **Owner** drop-down list defines the user to own this table.

The **Tablespace** drop-down list defines a tablespace different from the default tablespace to create a table.

**Note:** Support from PostgreSQL 8.0 or later.

### Inherits from

This option specifies a list of tables from which the new table automatically inherits all columns. Use of inheritance creates a persistent relationship between the new child table and its parent table(s). Schema modifications to the parent(s) normally propagate to children as well, and by default the data of the child table is included in scans of the parent(s).

To set the new table to be inherited from one or several existing tables, just simply click  to open the editor(s) for editing.

### Has Oids

Check this option if you want to specify whether rows of the new table should have OIDs (object identifiers) assigned to them.


### Fill Factor

The fillfactor for a table is a percentage between 10 and 100. 100 (complete packing) is the default. When a smaller fillfactor is specified, INSERT operations pack table pages only to the indicated percentage; the remaining space on each page is reserved for updating rows on that page. This gives UPDATE a chance to place the updated copy of a row on the same page as the original, which is more efficient than placing it on a different page. For a table whose entries are never updated, complete packing is the best choice, but in heavily updated tables smaller fillfactors are appropriate.

**Note:** Support from PostgreSQL 8.2 or later.



## PostgreSQL Views


Views are useful for allowing users to access a set of relations (tables) as if it were a single table, and limiting their access to just that. Views can also be used to restrict access to rows (a subset of a particular table).

Just simply click  to open an object pane for **View**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit, open and delete the selected view.

### Create View

To create a new view

- Select anywhere on the object pane.
- Click the  **New View** from the object pane toolbar.  
or
- Right-click and select  **New View** from the popup menu.
- Edit view properties on the appropriate tabs of the View Designer.

**Hint:** To create new view you can also right-click the Views node of the navigation pane and select the  **New View** from the popup menu.

To create a new view with the same properties as one of the existing views has (using drag and drop method)




**Apply to:** current database {same connection}

- Select the view(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen view(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created view(s) will be named as "viewname\_**copy**".




**Apply to:** different database {same connection}  
different database {different connection} (Data Transfer tool will be activated)

- Select the view(s) for copying in the object pane.
- Drag and drop the chosen view(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel

To create a new view with modification as one of the existing views

- Select the view for modifying in the navigation pane/object pane.
- Right-click and select the  **Design View** from the popup menu.  
or
- Click the  **Design View** from the object pane toolbar.
- Modify view properties on the appropriate tabs of the View Designer.
- Click  **Save As**.

To create a new view with loading from a SQL file

- Select anywhere on the object pane.
- Click the  **New View** from the object pane toolbar.  
or
- Right-click and select  **New View** from the popup menu.
- Click  **Load**.

## Create View Shortcut



To create a view shortcut

- Select the view for editing in the navigation pane/object pane.
- Right-click and select **Create Open View Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.

**Note:** This option is used to provide a convenient way for you to open your view for entering data directly (Grid View/Form View) without activating the main Navicat.

## Edit View

To edit the existing view (manage its SQL definition etc)



- Select the view for editing in the navigation pane/object pane.
- Right-click and select the  **Design View** from the popup menu.  
or
- Click the  **Design View** from the object pane toolbar.
- Edit view properties on the appropriate tabs of the View Designer.

To change the name of the view

- Select the view for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.



## Open View

To open a view (manage view data)

- Select the view for opening in the navigation pane/object pane.
- Right-click and select the  **Open View** from the popup menu or simply double-click the view.  
or
- Click the  **Open View** from the object pane toolbar.

## Delete View

To delete a view

- Select the view for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete View** from the popup menu.  
or
- Click the  **Delete View** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve View Information

To achieve a view information

- Select the view in the object pane.
- Right-click the selected view and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.



## PostgreSQL View Designer

**View Designer** is the basic Navicat tool for working with views. It allows you to create new view and edit the existing view definition (view name and the SELECT statement it implements).

- [Working with View Builder](#)
- [Editing View SQL Definition](#)
- [Setting Advanced View Properties](#)
- Editing View Comment
- View SQL Preview
- [View Preview](#)
- [View Explain](#)

## **Working with PostgreSQL View Builder (Available only in Full Version)**

**View Builder** allows you to build views visually. It allows you to create and edit views without knowledge of SQL. See Query Builder for details.

## Editing PostgreSQL View SQL Definition

The **Definition** tab allows you to edit the view definition as SQL statement (SELECT statement it implements).

Example:

```
SELECT
    report.clients.RecordID
FROM
    report.clients
```


**Hint:** To customize the view of the editor and find out more features for sql editing, see [Editor View and More Features](#).

## Setting Advanced PostgreSQL View Properties

### Owner


The owner of the view.

## PostgreSQL View Preview



To preview the result of the view, click  **Preview** on the toolbar. If the query statement is correct, the **Result** and **Message** tabs will be opened.

The **Result** tab displays the data of the view as a grid and the **Message** tab displays the message log.

## PostgreSQL View Explain

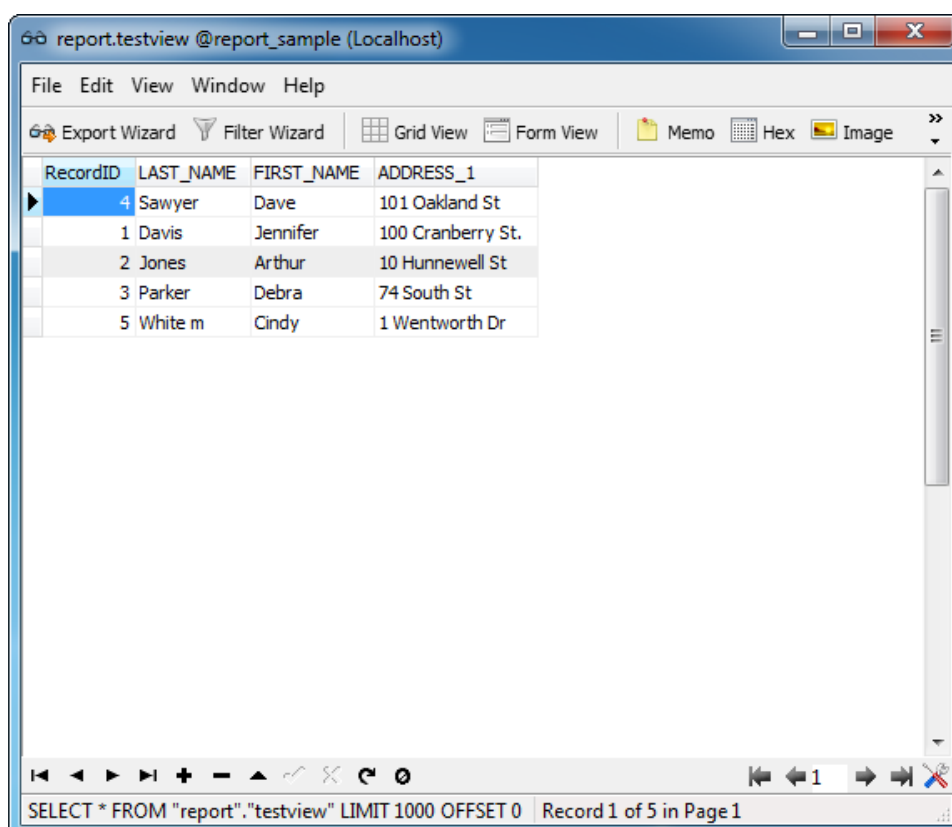
To show the Query Plan of the view, click  **Explain** on the toolbar. If the query statement is correct, the **Message** tab will show the query plan.

## PostgreSQL View Viewer

**View Viewer** displays the view data as a grid. Data can be displayed in three modes:  **Grid View**,  **Form View** and **Text/Blob View**. See Data View for details.

The toolbars of View Viewer provides the following functions for managing data:

- **Export Data**  
Export data to MS Word, MS Excel, MS Access, TXT, DBF, HTML, SQL, RTF and more.
- **Filter Data**  
Allow you to filter records by creating and applying filter criteria for the data grid.
- **Edit TEXT/BLOB**  
Allow you to view and edit the content of TEXT and BLOB fields.




## PostgreSQL Functions

PostgreSQL provides four kinds of functions:



- query language functions (functions written in SQL)
- procedural language functions (functions written in, for example, PL/Tcl or PL/pgSQL)
- internal functions
- C-language functions


Every kind of function can take base types, composite types, or combinations of these as arguments (parameters). In addition, every kind of function can return a base type or a composite type. Many kinds of functions can take or return certain pseudo-types (such as polymorphic types), but the available facilities vary.

Just simply click  to open an object pane for **Function**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected function.

### Create Function

To create a new function

- Select anywhere on the object pane.
- Click the  **New Function** from the object pane toolbar.  
or
- Right-click and select  **New Function** from the popup menu.
- Edit function properties on the appropriate tabs of the Function Designer.

**Hint:** To create new function you can also right-click the Function node of the navigation pane and select the  **New Function** from the popup menu.



To create a new function with the same properties as one of the existing function has (using drag and drop method)

**Apply to:** current database {same connection}




- Select the function(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen function(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created function(s) will be named as "functionname\_**copy**".

**Apply to:** different database {same connection}

different database {different connection} (Data Transfer tool will be activated)



- Select the function(s) for copying in the object pane.
- Drag and drop the chosen function(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel

To create a new function with modification as one of the existing function

- Select the function for modifying in the navigation pane/object pane.
- Right-click and select the  **Design Function** from the popup menu or simply double-click the function.  
or
- Click the  **Design Function** from the object pane toolbar.
- Modify function properties on the appropriate tabs of the Function Designer.
- Click  **Save As**.

## Edit Function

To edit the existing function (manage its definition, advanced etc)



- Select the function for editing in the navigation pane/object pane.
- Right-click and select the  **Design Function** from the popup menu or simply double-click the function.  
or
- Click the  **Design Function** from the object pane toolbar.
- Edit function properties on the appropriate tabs of the Function Designer.

To change the name of the function


- Select the function for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.

## Run Function

To run a function in the navigation pane/object pane



- Select the function for executing in the navigation pane/object pane.
- Click the  **Execute Function** from the object pane toolbar.  
or
- Right-click and select  **Execute Function** from the popup menu.
- View/edit the returned data on the Result tab.

To run a function in the Function Designer

- Create a new function/open the existing function.
- Click  **Run**.
- View/edit the returned data on the Result tab.

## Delete Function

To delete a function


- Select the function for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Function** from the popup menu.  
or
- Click the  **Delete Function** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Function Information

To achieve a function information

- Select the function in the object pane.
- Right-click the selected function and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## PostgreSQL Function Wizard

Click the  **New Function** from the object pane toolbar. The **Function Wizard** will pop up and it allows you to create a function easily.

- [Setting Parameters for Function](#)
- [Setting Return Type for Function](#)

You are allowed not to show the **Function Wizard** when create new function.

**Hint:** Once uncheck the **Show wizard next time**, you can go to Options to enable it.

## Setting Parameters for PostgreSQL Function

### Function

Define the parameter(s) of the function. Set the parameter **Mode**, **Type Schema**, **Type**, **Name** and **Default Value** under corresponding columns.

## Setting Return Type for PostgreSQL Function

Select the **Schema** and **Return Type** from the list.

## PostgreSQL Function Designer

**Function Designer** is the basic Navicat tool for working with functions. It allows you to create new function and edit the existing function definition.

- [Editing Function Definition](#)
- [Setting Advanced Function Properties](#)
- Editing Function Comment
- Function SQL Preview
- [Viewing Function Result](#)

## Editing PostgreSQL Function Definition

Edit the function definition under the **Definition** tab. Definition consists of a valid SQL procedure statement. This can be a simple statement such as *SELECT* or *INSERT*, or it can be a compound statement written using *BEGIN* and *END*. Compound statements can contain declarations, loops, and other control structure statements. The general form of these statements follows.

Example:

```
BEGIN
    RETURN i + j;
END
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.

### Parameter

Defines function parameter.

### Return type schema and Return Type

It indicates the return type of the function.



## Setting Advanced PostgreSQL Function Properties

### Owner

The owner of the function.

**Note:** Support from PostgreSQL 8.0 or later.

### Language

The name of the language that the function is implemented in. May be SQL, C, internal, or the name of a user-defined procedural language. For backward compatibility, the name may be enclosed by single quotes.

### Volatility

These attributes inform the query optimizer about the behavior of the function. At most one choice may be specified. If none of these appear, VOLATILE is the default assumption.

**IMMUTABLE** indicates that the function cannot modify the database and always returns the same result when given the same argument values; that is, it does not do database lookups or otherwise use information not directly present in its argument list. If this option is given, any call of the function with all-constant arguments can be immediately replaced with the function value.

**STABLE** indicates that the function cannot modify the database, and that within a single table scan it will consistently return the same result for the same argument values, but that its result could change across SQL statements. This is the appropriate selection for functions whose results depend on database lookups, parameter variables (such as the current time zone), etc. Also note that the `current_timestamp` family of functions qualify as stable, since their values do not change within a transaction.

**VOLATILE** indicates that the function value can change even within a single table scan, so no optimizations can be made. Relatively few database functions are volatile in this sense; some examples are `random()`, `curval()`, `timeofday()`. But note that any function that has side-effects must be classified volatile, even if its result is quite predictable, to prevent calls from being optimized away; an example is `setval()`.

### ☒ Security of definer

Specifies that the function is to be executed with the privileges of the user that created it.

### ☒ Returns Set

Indicates that the function will return a set of items, rather than a single item.

## **Strict**

Indicates that the function always returns null whenever any of its arguments are null. If this parameter is specified, the function is not executed when there are null arguments; instead a null result is assumed automatically.

## **Estimated cost**

A positive number giving the estimated execution cost for the function, in units of `cpu_operator_cost`. If the function returns a set, this is the cost per returned row. If the cost is not specified, 1 unit is assumed for C-language and internal functions, and 100 units for functions in all other languages. Larger values cause the planner to try to avoid evaluating the function more often than necessary.

**Note:** Support from PostgreSQL 8.3 or later.

## **Estimated rows**

A positive number giving the estimated number of rows that the planner should expect the function to return. This is only allowed when the function is declared to return a set. The default assumption is 1000 rows.


**Note:** Support from PostgreSQL 8.3 or later.

## **Configuration parameter**

The specified configuration parameter to be set to the specified value when the function is entered, and then restored to its prior value when the function exits.

**Note:** Support from PostgreSQL 8.3 or later.

## Viewing PostgreSQL Function Result

To run the function click  **Run** on the toolbar. If the SQL statement is correct, the statement will be executed and, if the statement is supposed to return data, the **Result** tab opens with the data returned by the function. If an error occurs while executing the function, execution stops, the appropriate error message is displayed.


If the function requires input parameter, the **Input Parameters** box will popup. Use ',' to separate the parameters.

**Note:** The **Result** tab displays the result data as grid.

**Hint:** Navicat supports to return 10 resultsets.



## PostgreSQL Aggregates

Aggregate functions in PostgreSQL are expressed as state values and state transition functions. That is, an aggregate can be defined in terms of state that is modified whenever an input item is processed. To define a new aggregate function, one selects a data type for the state value, an initial value for the state, and a state transition function. The state transition function is just an ordinary function that could also be used outside the context of the aggregate. A final function can also be specified, in case the desired result of the aggregate is different from the data that needs to be kept in the running state value.




Just simply click  -> Aggregate to open an object pane for **Aggregate**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected aggregate.

### Create Aggregate

To create a new aggregate



- Select anywhere on the object pane.
- Click the  **New Aggregate** from the object pane toolbar.  
or
- Right-click and select  **New Aggregate** from the popup menu.
- Edit aggregate properties on the appropriate tabs of the Aggregate Designer.

To create a new aggregate with modification as one of the existing aggregate

- Select the aggregate for modifying in the object pane.
- Right-click and select the  **Design Aggregate** from the popup menu or simply double-click the aggregate.  
or
- Click the  **Design Aggregate** from the object pane toolbar.
- Modify aggregate properties on the appropriate tabs of the Aggregate Designer.
- Click  **Save As**.

## Edit Aggregate

To edit the existing aggregate (manage its properties etc)

- Select the aggregate for editing in the object pane.
- Right-click and select the  **Design Aggregate** from the popup menu or simply double-click the aggregate.  
or
- Click the  **Design Aggregate** from the object pane toolbar.
- Edit aggregate properties on the appropriate tabs of the Aggregate Designer.



To change the name of the aggregate

- Select the aggregate for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

**Note:** Support from PostgreSQL 7.4 or later.

## Delete Aggregate

To delete an aggregate

- Select the aggregate for deleting in the object pane.
- Right-click and select the  **Delete Aggregate** from the popup menu.  
or
- Click the  **Delete Aggregate** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Aggregate Information

To achieve an aggregate information

- Select the aggregate in the object pane.
- Right-click the selected aggregate and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## PostgreSQL Aggregate Designer

**Aggregate Designer** is the basic Navicat tool for working with aggregates. It allows you to create new aggregate and edit the existing aggregate properties.

- [Editing Aggregate Properties](#)
- Editing Aggregate Comment
- Aggregate SQL Preview

## Editing PostgreSQL Aggregate Properties

### Owner

The owner of the aggregate function.

**Note:** Support from PostgreSQL 8.0 or later.

### Input type

An input data type on which this aggregate function operates.

**Note:** Support from PostgreSQL 8.2 or later. For versions below 8.2, just select the **Input type schema** and **Input type** from the dropdown lists.

### State type schema and State type

The data type for the aggregate's state value.

### State function schema and State function

The state transition function to be called for each input row. For an N-argument aggregate function, the state function must take N+1 arguments, the first being of type *state\_data\_type* and the rest matching the declared input data type(s) of the aggregate. The function must return a value of type *state\_data\_type*. This function takes the current state value and the current input data value(s), and returns the next state value.

### Final function schema and Final function

The final function called to compute the aggregate's result after all input rows have been traversed. The function must take a single argument of type *state\_data\_type*. The return data type of the aggregate is defined as the return type of this function. If final function is not specified, then the ending state value is used as the aggregate's result, and the return type is *state\_data\_type*.

### Initial condition

The initial setting for the state value. This must be a string constant in the form accepted for the data type *state\_data\_type*. If not specified, the state value starts out null.


### Sort operator schema and Sort operator

The associated sort operator for a MIN- or MAX-like aggregate. The operator is assumed to have the same input data types as the aggregate (which must be a single-argument aggregate).

**Note:** Support from PostgreSQL 8.1 or later.



## PostgreSQL Conversions

Conversion defines a new conversion between character set encodings. Conversion names may be used in the convert function to specify a particular encoding conversion. Also, conversions that are marked DEFAULT can be used for automatic encoding conversion between client and server. For this purpose, two conversions, from encoding A to B and from encoding B to A, must be defined.




Just simply click  -> Conversion to open an object pane for **Conversion**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected conversion.

### Create Conversion

To create a new conversion

- Select anywhere on the object pane.
- Click the  **New Conversion** from the object pane toolbar.  
or
- Right-click and select  **New Conversion** from the popup menu.
- Edit conversion properties on the appropriate tabs of the Conversion Designer.



To create a new conversion with modification as one of the existing conversion

- Select the conversion for modifying in the object pane.
- Right-click and select the  **Design Conversion** from the popup menu or simply double-click the conversion.  
or
- Click the  **Design Conversion** from the object pane toolbar.
- Modify conversion properties on the appropriate tabs of the Conversion Designer.
- Click  **Save As**.



## Edit Conversion

To edit the existing conversion (manage its properties etc)

- Select the conversion for editing in the object pane.
- Right-click and select the  **Design Conversion** from the popup menu or simply double-click the conversion.  
or
- Click the  **Design Conversion** from the object pane toolbar.
- Edit conversion properties on the appropriate tabs of the Conversion Designer.



To change the name of the conversion

- Select the conversion for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

**Note:** Support from PostgreSQL 7.4 or later.

## Delete Conversion

To delete a conversion

- Select the conversion for deleting in the object pane.
- Right-click and select the  **Delete Conversion** from the popup menu.  
or
- Click the  **Delete Conversion** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Conversion Information

To achieve a conversion information

- Select the conversion in the object pane.
- Right-click the selected conversion and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## PostgreSQL Conversion Designer

**Conversion Designer** is the basic Navicat tool for working with conversions. It allows you to create new conversion and edit the existing conversion properties.

- [Editing Conversion Properties](#)
- Editing Conversion Comment
- Conversion SQL Preview

## Editing PostgreSQL Conversion Properties

### Owner

The owner of the conversion function.

**Note:** Support from PostgreSQL 8.0 or later.

### Source encoding

The source encoding name.

### Target encoding

The destination encoding name.

### Schema of function and Function

The function used to perform the conversion. The function name may be schema-qualified. If it is not, the function will be looked up in the path.

The function must have the following signature:

```
conv_proc(  
integer, -- source encoding ID  
integer, -- destination encoding ID  
cstring, -- source string (null terminated C string)  
internal, -- destination (fill with a null terminated C string)  
integer -- source string length  
) RETURNS void;
```


### ☒ Default

Check this box to indicate that this conversion is the default for this particular source to destination encoding. There should be only one default encoding in a schema for the encoding pair.

## PostgreSQL Domains



A domain is essentially a data type with optional constraints (restrictions on the allowed set of values). The user who defines a domain becomes its owner.

Domains are useful for abstracting common constraints on fields into a single location for maintenance. For example, several tables might contain email address columns, all requiring the same *CHECK* constraint to verify the address syntax. Define a domain rather than setting up each table's constraint individually.




Just simply click  -> Domain to open an object pane for **Domain**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected domain.

### Create Domain

To create a new domain



- Select anywhere on the object pane.
- Click the  **New Domain** from the object pane toolbar.  
or
- Right-click and select  **New Domain** from the popup menu.
- Edit domain properties on the appropriate tabs of the Domain Designer.

To create a new domain with modification as one of the existing domain

- Select the domain for modifying in the object pane.
- Right-click and select the  **Design Domain** from the popup menu or simply double-click the domain.  
or
- Click the  **Design Domain** from the object pane toolbar.
- Modify domain properties on the appropriate tabs of the Domain Designer.
- Click  **Save As**.

## Edit Domain

To edit the existing domain (manage its general etc)



- Select the domain for editing in the object pane.
- Right-click and select the  **Design Domain** from the popup menu or simply double-click the domain.  
or
- Click the  **Design Domain** from the object pane toolbar.
- Edit domain properties on the appropriate tabs of the Domain Designer.

To change the name of the domain

- Select the domain for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

## Delete Domain

To delete a domain

- Select the domain for deleting in the object pane.
- Right-click and select the  **Delete Domain** from the popup menu.  
or
- Click the  **Delete Domain** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Domain Information

To achieve a domain information

- Select the domain in the object pane.
- Right-click the selected domain and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## PostgreSQL Domain Designer

**Domain Designer** allows you to define domain properties and its checks. It allows you to create new domain and edit the existing domain properties.

- [Editing Domain General](#)
- [Editing Domain Check](#)
- Editing Domain Comment
- Domain SQL Preview

## Editing PostgreSQL Domain General

### Underlying Type Category

Choose the underlying data type category: **Base Type**, **Composite Type**, **Enum Type** and **Domain**.

**Note:** Support from PostgreSQL 8.2 or later.

### Underlying Type Schema

Select schema of the underlying data type.

### Underlying Type

Select the underlying data type of the domain from the drop-down list.

### Dimensions

The dimensions of array specifiers.

### Length and Scale

Use the **Length** edit box to define the length of the field and use **Scale** edit box to define the number of digits after the decimal point. (if required for the selected data type)

### Default

The *DEFAULT* clause specifies a default value for columns of the domain data type. The value is any variable-free expression (but subqueries are not allowed). The data type of the default expression must match the data type of the domain. If no default value is specified, then the default value is the null value.

The default expression will be used in any insert operation that does not specify a value for the column. If a default value is defined for a particular column, it overrides any default associated with the domain. In turn, the domain default overrides any default value associated with the underlying data type.

### ☒ Not null

Values of this domain are not allowed to be null.

### Owner

The owner of the domain function. The user who defines a domain becomes its owner.

**Note:** Support from PostgreSQL 7.4 or later.

## Editing PostgreSQL Domain Check

The **Checks** tab is provided for managing domain checks. It allows you to create new, edit, or delete the selected check.

*CHECK* clauses specify integrity constraints or tests which values of the domain must satisfy. Each constraint must be an expression producing a Boolean result. It should use the key word *VALUE* to refer to the value being tested.

See [Checks](#) for details.




## PostgreSQL Trigger Functions

Trigger Function can be created with PL/pgSQL and referenced within a PostgreSQL trigger definition. The term "trigger function" is simply a way of referring to a function that is intended to be invoked by a trigger. Triggers define operations that are performed when a specific event occurs within the database. A PL/pgSQL trigger function can be referenced by a trigger as the operation to be performed when the trigger's event occurs.



The definition of a trigger and the definition of its associated trigger function are two different things. A trigger is defined with the SQL *CREATE TRIGGER* command, whereas trigger functions are defined using the SQL *CREATE FUNCTION* command.

See [Triggers](#) for details.




Just simply click  -> Trigger Function to open an object pane for **Trigger Function**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected trigger function.

### Create Trigger Function

To create a new trigger function



- Select anywhere on the object pane.
- Click the  **New Trigger Function** from the object pane toolbar.  
or
- Right-click and select  **New Trigger Function** from the popup menu.
- Edit trigger function properties on the appropriate tabs of the Trigger Function Designer.

To create a new trigger function with modification as one of the existing trigger function

- Select the trigger function for modifying in the object pane.
- Right-click and select the  **Design Trigger Function** from the popup menu or simply double-click the trigger function.  
or
- Click the  **Design Trigger Function** from the object pane toolbar.
- Modify trigger function properties on the appropriate tabs of the Trigger Function Designer.
- Click  **Save As**.

## Edit Trigger Function

To edit the existing trigger function(manage its definition, advanced, etc)



- Select the trigger function for editing in the object pane.
- Right-click and select the  **Design Trigger Function** from the popup menu or simply double-click the trigger function.  
or
- Click the  **Design Trigger Function** from the object pane toolbar.
- Edit trigger function properties on the appropriate tabs of the Trigger Function Designer.

To change the name of the trigger function

- Select the trigger function for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

## Delete Trigger Function

To delete a trigger function

- Select the trigger function for deleting in the object pane.
- Right-click and select the  **Delete Trigger Function** from the popup menu.  
or
- Click the  **Delete Trigger Function** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Trigger Function Information

To achieve a trigger function information

- Select the trigger function in the object pane.
- Right-click the selected trigger function and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## PostgreSQL Trigger Function Designer

**Trigger Function Designer** is the basic Navicat tool for working with trigger functions. It allows you to create new trigger function and edit the existing trigger function definition.

- [Editing Trigger Function Definition](#)
- [Setting Advanced Trigger Function Properties](#)
- Editing Trigger Function Comment
- Trigger Function SQL Preview

## Editing PostgreSQL Trigger Function Definition

Edit the trigger function definition under the **Definition** tab. Definition consists of a valid SQL procedure statement. This can be a simple statement such as *SELECT* or *INSERT*, or it can be a compound statement written using *BEGIN* and *END*. Compound statements can contain declarations, loops, and other control structure statements.

### **Parameter**

Defines trigger function parameter.

### **Return type schema and Return Type**

It indicates the return type of the trigger function.

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.

## Setting Advanced PostgreSQL Trigger Function Properties

### Owner

The owner of the trigger function.

**Note:** Support from PostgreSQL 8.0 or later.

### Language

The name of the language that the function is implemented in. May be C, internal, or the name of a user-defined procedural language. For backward compatibility, the name may be enclosed by single quotes.

### Volatility

These attributes inform the query optimizer about the behavior of the function. At most one choice may be specified. If none of these appear, VOLATILE is the default assumption.

**IMMUTABLE** indicates that the function cannot modify the database and always returns the same result when given the same argument values; that is, it does not do database lookups or otherwise use information not directly present in its argument list. If this option is given, any call of the function with all-constant arguments can be immediately replaced with the function value.

**STABLE** indicates that the function cannot modify the database, and that within a single table scan it will consistently return the same result for the same argument values, but that its result could change across SQL statements. This is the appropriate selection for functions whose results depend on database lookups, parameter variables (such as the current time zone), etc. Also note that the `current_timestamp` family of functions qualify as stable, since their values do not change within a transaction.

**VOLATILE** indicates that the function value can change even within a single table scan, so no optimizations can be made. Relatively few database functions are volatile in this sense; some examples are `random()`, `curval()`, `timeofday()`. But note that any function that has side-effects must be classified volatile, even if its result is quite predictable, to prevent calls from being optimized away; an example is `setval()`.

### ☒ Security of definer

Specifies that the function is to be executed with the privileges of the user that created it.

### ☒ Returns Set

Indicates that the function will return a set of items, rather than a single item.

## **Strict**

Indicates that the function always returns null whenever any of its arguments are null. If this parameter is specified, the function is not executed when there are null arguments; instead a null result is assumed automatically.

## **Estimated cost**

A positive number giving the estimated execution cost for the function, in units of `cpu_operator_cost`. If the function returns a set, this is the cost per returned row. If the cost is not specified, 1 unit is assumed for C-language and internal functions, and 100 units for functions in all other languages. Larger values cause the planner to try to avoid evaluating the function more often than necessary.

**Note:** Support from PostgreSQL 8.2 or later.

## **Estimated rows**

A positive number giving the estimated number of rows that the planner should expect the function to return. This is only allowed when the function is declared to return a set. The default assumption is 1000 rows.

**Note:** Support from PostgreSQL 8.2 or later.

## **Configuration parameter**

The specified configuration parameter to be set to the specified value when the function is entered, and then restored to its prior value when the function exits.


**Note:** Support from PostgreSQL 8.2 or later.

## PostgreSQL Operators

PostgreSQL supports left unary, right unary, and binary operators. Operators can be overloaded.



At least one of *LEFTARG* and *RIGHTARG* must be defined. For binary operators, both must be defined. For right unary operators, only *LEFTARG* should be defined, while for left unary operators only *RIGHTARG* should be defined.

**Note:** *LEFTARG* = Left type; *RIGHTARG* = Right type.




Just simply click  -> Operator to open an object pane for **Operator**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected operator.

### Create Operator

To create a new operator



- Select anywhere on the object pane.
- Click the  **New Operator** from the object pane toolbar.  
or
- Right-click and select  **New Operator** from the popup menu.
- Edit operator properties on the appropriate tabs of the Operator Designer.

To create a new operator with modification as one of the existing operator

- Select the operator for modifying in the object pane.
- Right-click and select the  **Design Operator** from the popup menu or simply double-click the operator.  
or
- Click the  **Design Operator** from the object pane toolbar.
- Modify operator properties on the appropriate tabs of the Operator Designer.
- Click  **Save As**.

## Edit Operator

To edit the existing operator (manage its general etc)



- Select the operator for editing in the object pane.
- Right-click and select the  **Design Operator** from the popup menu or simply double-click the operator.  
or
- Click the  **Design Operator** from the object pane toolbar.
- Edit operator properties on the appropriate tabs of the Operator Designer.

To change the name of the operator

- Select the operator for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

## Delete Operator

To delete an operator

- Select the operator for deleting in the object pane.
- Right-click and select the  **Delete Operator** from the popup menu.  
or
- Click the  **Delete Operator** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Operator Information

To achieve an operator information

- Select the operator in the object pane.
- Right-click the selected operator and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.



## PostgreSQL Operator Designer

**Operator Designer** is the basic Navicat tool for working with operator. It allows you to create new operator and edit the existing operator properties.

- [Editing Operator General](#)
- [Editing Advanced Operator Properties](#)
- Editing Operator Comment
- Operator SQL Preview

## Editing PostgreSQL Operator General

### Owner

The owner of the operator function.

**Note:** Support from PostgreSQL 8.0 or later.

### Schema of left type and Left type

The data type of the operator's left operand, if any. This option would be omitted for a left-unary operator.

### Schema of right type and Right type

The data type of the operator's right operand, if any. This option would be omitted for a right-unary operator.

### Schema of operator function and Operator function

The function used to implement this operator.

## Editing Advanced PostgreSQL Operator Properties

### Schema of restrict function and Restrict function

The restriction selectivity estimator function for this operator.

### Schema of join function and Join function

The join selectivity estimator function for this operator.

### Schema of commutator and Commutator

The commutator of this operator.

### Schema of negator and Negator

The negator of this operator.

#### ☒ Hash

The operator can support a hash join if this option on.

#### ☒ Merge

The operator can support a merge join if this option on.

## Additional information for PostgreSQL version below 8.3

### Schema of left sort operator and Left sort operator

If this operator can support a merge join, the left sort operator that sorts the left-hand data type of this operator.

### Schema of right sort operator and Right sort operator

If this operator can support a merge join, the right sort operator that sorts the right-hand data type of this operator.

### Schema of less than operator and Less than operator

If this operator can support a merge join, the less-than operator that compares the input data types of this operator.


### Schema of greater than operator and Greater than operator

If this operator can support a merge join, the greater-than operator that compares the input data types of this operator.

## PostgreSQL Operator Classes



An operator class defines how a particular data type can be used with an index. The operator class specifies that certain operators will fill particular roles or "strategies" for this data type and this index method. The operator class also specifies the support procedures to be used by the index method when the operator class is selected for an index column. All the operators and functions used by an operator class must be defined before the operator class is created.

**Note:** Two operator classes in the same schema can have the same name only if they are for different index methods.




Just simply click  -> Operator Class to open an object pane for **Operator Class**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected operator class.

### Create Operator Class

To create a new operator class



- Select anywhere on the object pane.
- Click the  **New Operator Class** from the object pane toolbar.  
or
- Right-click and select  **New Operator Class** from the popup menu.
- Edit operator class properties on the appropriate tabs of the Operator Class Designer.

To create a new operator class with modification as one of the existing operator class

- Select the operator class for modifying in the object pane.
- Right-click and select the  **Design Operator Class** from the popup menu or simply double-click the operator class.  
or
- Click the  **Design Operator Class** from the object pane toolbar.
- Modify operator class properties on the appropriate tabs of the Operator Class Designer.
- Click  **Save As**.

## Edit Operator Class

To edit the existing operator class(manage its general, operators etc)



- Select the operator class for editing in the object pane.
- Right-click and select the  **Design Operator Class** from the popup menu or simply double-click the operator class.  
or
- Click the  **Design Operator Class** from the object pane toolbar.
- Edit operator class properties on the appropriate tabs of the Operator Class Designer.

To change the name of the operator class

- Select the operator class for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

## Delete Operator Class

To delete an operator class

- Select the operator class for deleting in the object pane.
- Right-click and select the  **Delete Operator Class** from the popup menu.  
or
- Click the  **Delete Operator Class** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Operator Class Information

To achieve an operator class information

- Select the operator class in the object pane.
- Right-click the selected operator class and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## PostgreSQL Operator Class Designer

**Operator Class Designer** is the basic Navicat tool for working with operator class. It allows you to create new operator class and edit the existing operator class properties.

- [Editing Operator Class General](#)
- [Editing Operator Class Operators](#)
- [Editing Operator Class Functions](#)
- Editing Operator Class Comment (Support from PostgreSQL 8.0 or later)
- Operator Class SQL Preview

## Editing PostgreSQL Operator Class General

### Owner

The owner of the operator class function.

**Note:** Support from PostgreSQL 8.0 or later.

### Schema of data type and Data Type

The column data type that this operator class is for.

### Index method

The name of the index method this operator class is for.

### Schema of storage type and Storage type

The data type actually stored in the index. Normally this is the same as the column data type, but some index methods (*GIN* and *GiST* for now) allow it to be different. The *STORAGE* clause must be omitted unless the index method allows a different type to be used.

### Operator family

The name of the existing operator family to add this operator class to. If not specified, a family named the same as the operator class is used (creating it, if it doesn't already exist).

**Note:** Support from PostgreSQL 8.3 or later.

### ☒ Default operator class

With this option selected, the operator class will become the default operator class for its data type. At most one operator class can be the default for a specific data type and index method.

## Editing PostgreSQL Operator Class Operators

### Strategy number

The index method's strategy number for an operator associated with the operator class.

### Schema of operator and Operator name

The operator associated with the operator class.

### ☒ Recheck

With this option selected, the index is "lossy" for this operator, and so the rows retrieved using the index must be rechecked to verify that they actually satisfy the qualification clause involving this operator.

**Note:** Before PostgreSQL 8.4, the OPERATOR clause could include a RECHECK option. This is no longer supported because whether an index operator is "lossy" is now determined on-the-fly at runtime. This allows efficient handling of cases where an operator might or might not be lossy.



## Editing PostgreSQL Operator Class Functions

### Support number


The index method's support procedure number for a function associated with the operator class.

### Schema of function and Function name

The function that is an index method support procedure for the operator class.



## PostgreSQL Sequences

Sequence involves creating and initializing a new special single-row table. It is usually used to generate unique identifiers for rows of a table.




Just simply click  -> Sequence to open an object pane for **Sequence**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected sequence.

### Create Sequence

To create a new sequence



- Select anywhere on the object pane.
- Click the  **New Sequence** from the object pane toolbar.  
or
- Right-click and select  **New Sequence** from the popup menu.
- Edit sequence properties on the appropriate tabs of the Sequence Designer.

To create a new sequence with modification as one of the existing sequence

- Select the sequence for modifying in the object pane.
- Right-click and select the  **Design Sequence** from the popup menu or simply double-click the sequence.  
or
- Click the  **Design Sequence** from the object pane toolbar.
- Modify sequence properties on the appropriate tabs of the Sequence Designer.
- Click  **Save As**.

### Edit Sequence

To edit the existing sequence(manage its general etc)



- Select the sequence for editing in the object pane.
- Right-click and select the  **Design Sequence** from the popup menu or simply double-click the sequence.  
or
- Click the  **Design Sequence** from the object pane toolbar.
- Edit sequence properties on the appropriate tabs of the Sequence Designer.

To change the name of the sequence

- Select the sequence for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

## Delete Sequence

To delete a sequence

- Select the sequence for deleting in the object pane.
- Right-click and select the  **Delete Sequence** from the popup menu.  
or
- Click the  **Delete Sequence** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Sequence Information

To achieve a sequence information

- Select the sequence in the object pane.
- Right-click the selected sequence and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## PostgreSQL Sequence Designer

**Sequence Designer** is the basic Navicat tool for working with sequence. It allows you to create new sequence and edit the existing sequence properties.

- [Editing Sequence General](#)
- Editing Sequence Comment
- Sequence SQL Preview

## Editing PostgreSQL Sequence General

### Owner

The owner of the sequence function.

**Note:** Support from PostgreSQL 8.0 or later.

### Increment

Specifies which value is added to the current sequence value to create a new value. A positive value will make an ascending sequence, a negative one a descending sequence. The default value is 1.

### Current value

The starting value of the sequence.

### Minimum

Determines the minimum value a sequence can generate. If no minimum value is specified, then defaults will be used.

### Maximum

Determines the maximum value for the sequence. If no maximum value is specified, then default values will be used.

### Cache

Specifies how many sequence numbers are to be preallocated and stored in memory for faster access. The minimum value is 1 (only one value can be generated at a time, i.e., no cache), and this is also the default.

### ☒ Cycled

This option allows the sequence to wrap around when the maxvalue or minvalue has been reached by an ascending or descending sequence respectively. If the limit is reached, the next number generated will be the minvalue maxvalue, respectively. Otherwise, any calls to nextval after the sequence has reached its maximum value will return an error.


## ☒ **Add owned by**

Choose the **Owned by table** and **Owned by column** so that the sequence is associated with a specific table column, such that if that column (or its whole table) is dropped, the sequence will be automatically dropped as well. The specified table must have the same owner and be in the same schema as the sequence.

**Note:** Support from PostgreSQL 8.2 or later.

## PostgreSQL Types




Type registers a new data type for use in the current database. If a schema name is given then the type is created in the specified schema. Otherwise it is created in the current schema. The type name must be distinct from the name of any existing type or domain in the same schema. (Because tables have associated data types, the type name must also be distinct from the name of any existing table in the same schema.)

Just simply click  -> Type to open an object pane for **Type**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected type.




**Note:** Enum Type was added in PostgreSQL 8.3.

### Create Type

To create a new type



- Select anywhere on the object pane.
- Click the  **New Type** from the object pane toolbar together with the  down arrow to choose **New Base Type** / **New Composite Type** / **New Enum Type**.  
or
- Right-click and select  **New Type** -> **New Base Type** / **New Composite Type** / **New Enum Type** from the popup menu.
- Edit type properties on the appropriate tabs of the Type Designer.

To create a new type with modification as one of the existing type

- Select the type for modifying in the object pane.
- Right-click and select the  **Design Type** from the popup menu or simply double-click the type.  
or
- Click the  **Design Type** from the object pane toolbar.
- Modify type properties on the appropriate tabs of the Type Designer.
- Click  **Save As**.



## Edit Type

To edit the existing type(manage its general etc)

- Select the type for editing in the object pane.
- Right-click and select the  **Design Type** from the popup menu or simply double-click the type.  
or
- Click the  **Design Type** from the object pane toolbar.
- Edit type properties on the appropriate tabs of the Type Designer.

## Delete Type

To delete a type

- Select the type for deleting in the object pane.
- Right-click and select the  **Delete Type** from the popup menu.  
or
- Click the  **Delete Type** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Type Information

To achieve a type information

- Select the type in the object pane.
- Right-click the selected type and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.



## PostgreSQL Type Designer

**Type Designer** is the basic Navicat tool for working with type. It allows you to create new type and edit the existing type properties.

- [Editing Base Type Properties](#)
- [Editing Composite Type Properties](#)
- [Editing Enum Type Properties](#)
- Editing Type Comment
- Type SQL Preview

## Editing PostgreSQL Base Type Properties

**Base types** are those, like int4, that are implemented below the level of the SQL language (typically in a low-level language such as C). They generally correspond to what are often known as abstract data types. PostgreSQL can only operate on such types through functions provided by the user and only understands the behavior of such types to the extent that the user describes them. Base types are further subdivided into scalar and array types. For each scalar type, a corresponding array type is automatically created that can hold variable-size arrays of that scalar type.

- [Editing Base Type General](#)
- [Editing Advanced Base Type Properties](#)

## Editing PostgreSQL Base Type General

### Input Schema and Input

The function that converts data from the type's external textual form to its internal form.

### Output Schema and Output

The function that converts data from the type's internal form to its external textual form.

### Length

A numeric constant that specifies the length in bytes of the new type's internal representation. The default assumption is that it is variable-length.

#### ☒ Variable

Checks this option if the type length is unknown.

### Default

The default value for the data type. If this is omitted, the default is null.

### Element

The type being created is an array; this specifies the type of the array elements.

### Delimiter

The delimiter character to be used between values in arrays made of this type.

### Alignment

The storage alignment requirement of the data type. If specified, it must be char, int2, int4, or double; the default is int4.

### Storage

The storage strategy for the data type. If specified, must be plain, external, extended, or main; the default is plain.

#### ☒ Pass by value

Indicates that values of this data type are passed by value rather than by reference.

### Owner

The owner of the type.

**Note:** Support from PostgreSQL 8.0 or later.

## Editing Advanced PostgreSQL Base Type Properties

The **Advanced** tab is supported from PostgreSQL 7.4 or later.

### Receive Schema and Receive

The function that converts data from the type's external binary form to its internal form.

### Send Schema and Send

The function that converts data from the type's internal form to its external binary form.

### Analyze Schema and Analyze

The function that performs statistical analysis for the data type.

**Note:** Support from PostgreSQL 8.0 or later.

### Type Modifier Input Schema and Type Modifier Input

The function that converts an array of modifier(s) for the type into internal form.

**Note:** Support from PostgreSQL 8.3 or later.

### Type Modifier Output Schema and Type Modifier Output

The function that converts the internal form of the type's modifier(s) to external textual form.

**Note:** Support from PostgreSQL 8.3 or later.

## Editing PostgreSQL Composite Type Properties

**Composite types**, or row types, are created whenever the user creates a table; it's also possible to define a "stand-alone" composite type with no associated table. A composite type is simply a list of base types with associated field names. A value of a composite type is a row or record of field values. The user can access the component fields from SQL queries.

- [Editing Composite Type General](#)

## Editing PostgreSQL Composite Type General

### Name

The name of an attribute (column) for the composite type.

### Type

The name of an existing data type to become a column of the composite type.

### Length and Scale

Use the **Length** edit box to define the length of the field and use **Scale** edit box to define the number of digits after the decimal point. (if required for the selected data type)

### Dimensions

The dimensions of array specifiers.

### Owner

The owner of the type.

**Note:** Support from PostgreSQL 8.0 or later.

## Editing PostgreSQL Enum Type Properties

**Enumerated (Enum) types** are data types that are comprised of a static, predefined set of values with a specific order. They are equivalent to the enum types in a number of programming languages. An example of an enum type might be the days of the week, or a set of status values for a piece of data.

**Note:** Enum Type was added in PostgreSQL 8.3.

- [Editing Enum Type General](#)

## Editing PostgreSQL Enum Type General

### Label

A string literal representing the textual label associated with one value of an enum type.


### Owner

The owner of the type.



## PostgreSQL Tablespaces



A tablespace allows superusers to define an alternative location on the file system where the data files containing database objects (such as tables and indexes) may reside.

Just simply click  -> Tablespace to open an object pane for **Tablespace**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected tablespace.

**Note:** Tablespace was added in PostgreSQL 8.0.



### Create Tablespace

To create a new tablespace

- Select anywhere on the object pane.
- Click the  **New Tablespace** from the object pane toolbar.  
or
- Right-click and select  **New Tablespace** from the popup menu.
- Edit tablespace properties on the appropriate tabs of the Tablespace Designer.

### Edit Tablespace

To edit the existing tablespace(manage its general, privileges etc)



- Select the tablespace for editing in the object pane.
- Right-click and select the  **Design Tablespace** from the popup menu or simply double-click the tablespace.  
or
- Click the  **Design Tablespace** from the object pane toolbar.
- Edit tablespace properties on the appropriate tabs of the Tablespace Designer.

To change the name of the tablespace

- Select the tablespace for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

## Delete Tablespace

To delete a tablespace

- Select the tablespace for deleting in the object pane.
- Right-click and select the  **Delete Tablespace** from the popup menu.  
or
- Click the  **Delete Tablespace** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Tablespace Information

To achieve a tablespace information

- Select the tablespace in the object pane.
- Right-click the selected tablespace and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## PostgreSQL Tablespace Designer

**Tablespace Designer** is the basic Navicat tool for working with tablespace. It allows you to create new tablespace and edit the existing tablespace properties.

- [Editing Tablespace General](#)
- Editing Tablespace Comment (Support from PostgreSQL 8.2 or later)
- Tablespace SQL Preview

## Editing PostgreSQL Tablespace General

### Location


The directory that will be used for the tablespace. The directory must be empty and must be owned by the PostgreSQL system user. The directory must be specified by an absolute path name.

### Owner

The name of the user who will own the tablespace. If omitted, defaults to the user executing the command. Only superusers may create tablespaces, but they can assign ownership of tablespaces to non-superusers.



## PostgreSQL Casts

A cast specifies how to perform a conversion between two data types.




Just simply click -> Cast to open an object pane for **Cast**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected cast.

### Create Cast

To create a new cast


- Select anywhere on the object pane.
- Click the  **New Cast** from the object pane toolbar.  
or
- Right-click and select  **New Cast** from the popup menu.
- Edit cast properties on the appropriate tabs of the Cast Designer.

To create a new cast with modification as one of the existing cast

- Select the cast for modifying in the object pane.
- Right-click and select the  **Design Cast** from the popup menu or simply double-click the cast.  
or
- Click the  **Design Cast** from the object pane toolbar.
- Modify cast properties on the appropriate tabs of the Cast Designer.
- Click  **Save As**.



### Edit Cast

To edit the existing cast(manage its general etc)

- Select the cast for editing in the object pane.
- Right-click and select the  **Design Cast** from the popup menu or simply double-click the cast.  
or
- Click the  **Design Cast** from the object pane toolbar.
- Edit cast properties on the appropriate tabs of the Cast Designer.

## Delete Cast

To delete a cast

- Select the cast for deleting in the object pane.
- Right-click and select the  **Delete Cast** from the popup menu.  
or
- Click the  **Delete Cast** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Cast Information

To achieve a cast information

- Select the cast in the object pane.
- Right-click the selected cast and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## PostgreSQL Cast Designer

**Cast Designer** is the basic Navicat tool for working with cast. It allows you to create new cast and edit the existing cast properties.

- [Editing Cast General](#)
- Editing Cast Comment (Support from PostgreSQL 8.0 or later)
- Cast SQL Preview

## Editing PostgreSQL Cast General

### Schema of source type and Source type

The schema and name of the source data type of the cast.

### Schema of target type and Target type

The schema and name of the target data type of the cast.

### Schema of function and Function

The function used to perform the cast. The function name may be schema-qualified. If it is not, the function will be looked up in the schema search path. The function's result data type must match the target type of the cast.

If no function is specify, indicates that the source type and the target type are binary compatible, so no function is required to perform the cast.

#### ☒ **Implicit**

Indicates that the cast may be invoked implicitly in any context.


#### ☒ **Assignment**

Indicates that the cast can be invoked implicitly in assignment contexts.





## PostgreSQL Languages

Language can register a new procedural language with a PostgreSQL database. Subsequently, functions and trigger procedures can be defined in this new language. The user must have the PostgreSQL superuser privilege to register a new language.




Just simply click -> Language to open an object pane for **Language**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected Language.

### Create Language

To create a new language



- Select anywhere on the object pane.
- Click the  **New Language** from the object pane toolbar.  
or
- Right-click and select  **New Language** from the popup menu.
- Edit language properties on the appropriate tabs of the Language Designer.

To create a new language with modification as one of the existing language

- Select the language for modifying in the object pane.
- Right-click and select the  **Design Language** from the popup menu or simply double-click the language.  
or
- Click the  **Design Language** from the object pane toolbar.
- Modify language properties on the appropriate tabs of the Language Designer.
- Click  **Save As**.

### Edit Language

To edit the existing language(manage its properties, privileges etc)

- Select the language for editing in the object pane.
- Right-click and select the  **Design Language** from the popup menu or simply double-click the language.  
or
- Click the  **Design Language** from the object pane toolbar.
- Edit language properties on the appropriate tabs of the Language Designer.



To change the name of the language

- Select the language for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

**Note:** Support from PostgreSQL 7.4 or later.

## Delete Language

To delete a language

- Select the language for deleting in the object pane.
- Right-click and select the  **Delete Language** from the popup menu.  
or
- Click the  **Delete Language** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Language Information

To achieve a language information

- Select the language in the object pane.
- Right-click the selected language and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## PostgreSQL Language Designer

**Language Designer** is the basic Navicat tool for working with language. It allows you to create new language and edit the existing language properties.

- [Editing Language General](#)
- Editing Language Comment (Support from PostgreSQL 8.0 or later)
- Language SQL Preview

## Editing PostgreSQL Language General

### Owner

The owner of the language.

**Note:** Support from PostgreSQL 8.3 or later.

### Schema of handler and Handler

Call Handler is the name of a previously registered function that will be called to execute the procedural language functions. The call handler for a procedural language must be written in a compiled language such as C with version 1 call convention and registered with PostgreSQL as a function taking no arguments and returning the *language\_handler* type, a placeholder type that is simply used to identify the function as a call handler.

### Schema of validator and Validator

Validator function is the name of a previously registered function that will be called when a new function in the language is created, to validate the new function. If no validator function is specified, then a new function will not be checked when it is created. The validator function must take one argument of type oid, which will be the OID of the to-be-created function, and will typically return void.

A validator function would typically inspect the function body for syntactical correctness, but it can also look at other properties of the function, for example if the language cannot handle certain argument types. To signal an error, the validator function should use the ereport() function. The return value of the function is ignored.

### **Trusted**

Specifies that the call handler for the language is safe, that is, it does not offer an unprivileged user any functionality to bypass access restrictions. If this key word is omitted when registering the language, only users with the PostgreSQL superuser privilege can use this language to create new functions.


## SQLite Database Object Management

The following list contains the most common SQLite database objects supported by Navicat.

- [Tables](#)
- [Views](#)
- [Indexes](#)
- [Triggers](#)



## SQLite Tables


Relational databases use tables to store data. All operations on data are done on the tables themselves or produce another tables as the result. A table is a set of rows and columns, and their intersections are fields. From a general perspective, columns within a table describe the name and type of data that will be found by row for that column's fields. Rows within a table represent records composed of fields that are described from left to right by their corresponding column's name and type. Each field in a row is implicitly correlated with each other field in that row.

Just simply click  to open an object pane for **Table**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit, open and delete the selected table.

### Create Table

To create a new table

- Select anywhere on the object pane.
- Click the  **New Table** from the object pane toolbar.  
or
- Right-click and select  **New Table** from the popup menu.
- Edit table properties and fields on the appropriate tabs of the Table Designer.

**Hint:** To create new table you can also right-click the Tables node of the navigation pane and select the  **New Table** from the popup menu.

To create a new table with the same properties as one of the existing tables has (using popup menu)

**Apply to:** current database {same connection}

- Select the table(s) for copying in the navigation pane/object pane.
- Right-click and select the **Duplicate Table** from the popup menu.
- The newly created table(s) will be named as "tablename\_**copy**".

To create a new table with the same properties as one of the existing tables has (using drag and drop method)

**Apply to:** current database {same connection}




- Select the table(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen table(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created table(s) will be named as "tablename\_**copy**"

**Apply to:** different database {same connection}

different database {different connection (same or cross server type)} (Data Transfer tool will be activated)

- Select the table(s) for copying in the object pane.
- Drag and drop the chosen table(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel

To create a new table with modification as one of the existing tables

- Select the table for modifying in the navigation pane/object pane.
- Right-click and select the  **Design Table** from the popup menu.  
or
- Click the  **Design Table** from the object pane toolbar.
- Modify table properties and fields on the appropriate tabs of the Table Designer.
- Click  **Save As**.

## Create Table Shortcut



To create a table shortcut

- Select the table for editing in the navigation pane/object pane.
- Right-click and select **Create Open Table Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.

**Note:** This option is used to provide a convenient way for you to open your table for entering data directly (Grid View/Form View) without activating the main Navicat.

## Edit Table

To edit the existing table (manage its fields, indexes, foreign keys and triggers etc)



- Select the table for editing in the navigation pane/object pane.
- Right-click and select the  **Design Table** from the popup menu.  
or
- Click the  **Design Table** from the object pane toolbar.
- Edit table properties and fields on the appropriate tabs of the Table Designer.


To change the name of the table

- Select the table for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.

## Open Table (manage table data)


To open a table

- Select the table for opening in the navigation pane/object pane.
- Right-click and select the  **Open Table** from the popup menu or simply double-click the table.  
or
- Click the  **Open Table** from the object pane toolbar.

**Note:** This option is only applied if you do wish Navicat loads all your images while opening the table. To open the graphical table with faster performance, use  **Open Table (Quick)** below.



To open a table with graphical fields

- Select the table for opening in the navigation pane/object pane.
- Right-click and select the  **Open Table (Quick)** from the popup menu.

**Note:** Faster performance for opening the graphical table, as BLOB fields (images) will not be loaded until you click on the cell.



## Empty Table

To empty a table

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Empty Table** from the popup menu.

## Delete Table

To delete a table

- Select the table for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Table** from the popup menu.  
or
- Click the  **Delete Table** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Table Information

To achieve a table information

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQLite Table Designer

**Table Designer** is the basic Navicat tool for working with tables. It allows you to create, edit and drop table's fields, indexes, foreign keys, and much more.



- [Managing Table Fields](#)
- [Managing Table Indexes](#)
- [Managing Table Foreign Keys](#)
- [Managing Table Uniques](#)
- [Managing Table Checks](#)
- [Managing Table Triggers](#)
- [Managing Table Options](#)
- Table SQL Preview

## SQLite Table Fields

Table fields are managed on the **Fields** tab of the Table Designer. Just simply click a field for editing. A right-click displays the popup menu or using field toolbar, allowing you to create new, insert, move and drop the selected field.

### Add Field

To add a field to the table


- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click and select the  **Add Field** from the popup menu or click the  **Add Field** from the toolbar.
- Edit field properties.

To add a new field with modification as one of the existing fields

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Select field.
- Right-click and select the **Duplicate Field** from the popup menu.
- Edit field properties.

### Insert Field

To insert a field above an existing field





- Open the table in the Table Designer.
- Open the **Fields** tab.
- Select field.
- Right-click and select the  **Insert Field** from the popup menu or click the  **Insert Field** from the toolbar.
- Define field properties in the empty row.

## Edit Field

To edit the table field



- Open the table in the Table Designer.
- Open the **Fields** tab.
- Simply click on the field to edit.

To change the order of the table fields


- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click on the field to move and select the  **Move Up**/  **Move Down** from the popup menu or click the  **Move Up**/  **Move Down** from the toolbar.

## Delete Field

To delete the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click on the field to delete and select the  **Delete Field** from the popup menu or click the  **Delete Field** from the toolbar.
- Confirm deleting in the dialog window.

## Setting SQLite Table Field Properties

Name	Type	Length	Decimals	Allow Null	
► CustNo	INTEGER	0	0	<input type="checkbox"/>	 1
Company	TEXT	0	0	<input checked="" type="checkbox"/>	
Addr1	TEXT	0	0	<input checked="" type="checkbox"/>	
Addr2	TEXT	0	0	<input checked="" type="checkbox"/>	

### Name

The **Name** is a descriptive identifier for a field that can be up to 64 characters (letters or numbers) including spaces. The names should be descriptive enough that anyone can easily identify them when viewing or editing records. For example, LastName, FirstName, StreetAddress, or HomePhone.

Use the **Name** edit box to set the field name. Note that the name of the field must be unique among all the field names in the table.

### Type

The **Type** dropdown list defines the type (storage class) of the field data.

The following tables summarize each type:

Type	Description
INTEGER	The value is a signed integer, stored in 1, 2, 3, 4, 6, or 8 bytes depending on the magnitude of the value.
REAL	The value is a floating point value, stored as an 8-byte IEEE floating point number.
TEXT	The value is a text string, stored using the database encoding (UTF-8, UTF-16BE or UTF-16LE).
BLOB	The value is a blob of data, stored exactly as it was input.

### Note for SQLite Version 2:

1. *You can store any kind of data you want in any column of any table, regardless of the declared datatype of that column.*

[Click here](#) for detailed description on datatype in SQLite version 2.

## Note for SQLite Version 3:

1. *Storage class is slightly more general than a datatype. The INTEGER storage class, for example, includes 6 different integer datatypes of different lengths.*
2. *In order to maximize compatibility between SQLite and other database engines, SQLite supports the concept of "type affinity" on columns.*

[Click here](#) for detailed description on datatype, storage class and type affinity.

## Length and Decimals

Use the **Length** edit box to define the length of the field and use **Decimals** edit box to define the number of digits after the decimal point (the scale).

### ☒ **Allow Null**

Allow the NULL values for the field.

### **Primary Key**

A **Primary Key** is a single field or combination of fields that uniquely defines a record. None of the fields that are part of the primary key can contain a null value.

## Setting Other SQLite Table Field Properties

### Default

To set the default value for the field.

### Collation

To specify the text collating function to use when comparing text entries for the column. The built-in BINARY collating function is used by default.

#### **BINARY**

Compares string data using memcmp(), regardless of text encoding.

#### **NOCASE**

The same as binary, except the 26 upper case characters of ASCII are folded to their lower case equivalents before the comparison is performed. Note that only ASCII characters are case folded. SQLite does not attempt to do full UTF case folding due to the size of the tables required.

#### **RTRIM**

The same as binary, except that trailing space characters are ignored.

**Note:** Support in SQLite 3.

### **Not null ON CONFLICT**

To specify an algorithm used to resolve constraint conflicts if Allow Null option is unchecked. The default conflict resolution algorithm is ABORT.

#### **ROLLBACK**

When a constraint violation occurs, an immediate ROLLBACK occurs, thus ending the current transaction, and the command aborts with a return code of SQLITE\_CONSTRAINT. If no transaction is active (other than the implied transaction that is created on every command) then this algorithm works the same as ABORT.

#### **ABORT**

When a constraint violation occurs, the command backs out any prior changes it might have made and aborts with a return code of SQLITE\_CONSTRAINT. But no ROLLBACK is executed so changes from prior commands within the same transaction are preserved. This is the default behavior.

## **FAIL**

When a constraint violation occurs, the command aborts with a return code `SQLITE_CONSTRAINT`. But any changes to the database that the command made prior to encountering the constraint violation are preserved and are not backed out. For example, if an `UPDATE` statement encountered a constraint violation on the 100th row that it attempts to update, then the first 99 row changes are preserved but changes to rows 100 and beyond never occur.

## **IGNORE**

When a constraint violation occurs, the one row that contains the constraint violation is not inserted or changed. But the command continues executing normally. Other rows before and after the row that contained the constraint violation continue to be inserted or updated normally. No error is returned when the `IGNORE` conflict resolution algorithm is used.

## **REPLACE**

When a `UNIQUE` constraint violation occurs, the pre-existing rows that are causing the constraint violation are removed prior to inserting or updating the current row. Thus the insert or update always occurs. The command continues executing normally following `REPLACE`. No error is returned by the `REPLACE` conflict resolution. If a `NOT NULL` constraint violation occurs, the `NULL` value is replaced by the default value for that column. If the column has no default value, then the `ABORT` algorithm is used. If a `CHECK` constraint violation occurs then the `IGNORE` algorithm is used.

### **Auto Increment** (INTEGER only)

The `AUTO INCREMENT` attribute can be used to generate a unique identity for new rows. To start with the `AUTO INCREMENT` value other than 1, you can set that value in Options tab.





## SQLite Table Indexes

Index provides a faster access path to table data. It is created using one or more columns of a table to speed SQL statement execution on that table.

Table indexes are managed on the **Indexes** tab of the Table Designer. Just simply click/double-click an index field for editing. A right-click displays the popup menu or using the index toolbar, allowing you to create new, edit and delete the selected index field.

### Add Index

To add a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Right-click and select the  **Add Index** from the popup menu or click the  **Add Index** from the toolbar.
- Edit index properties.



### Edit Index

To edit a table index

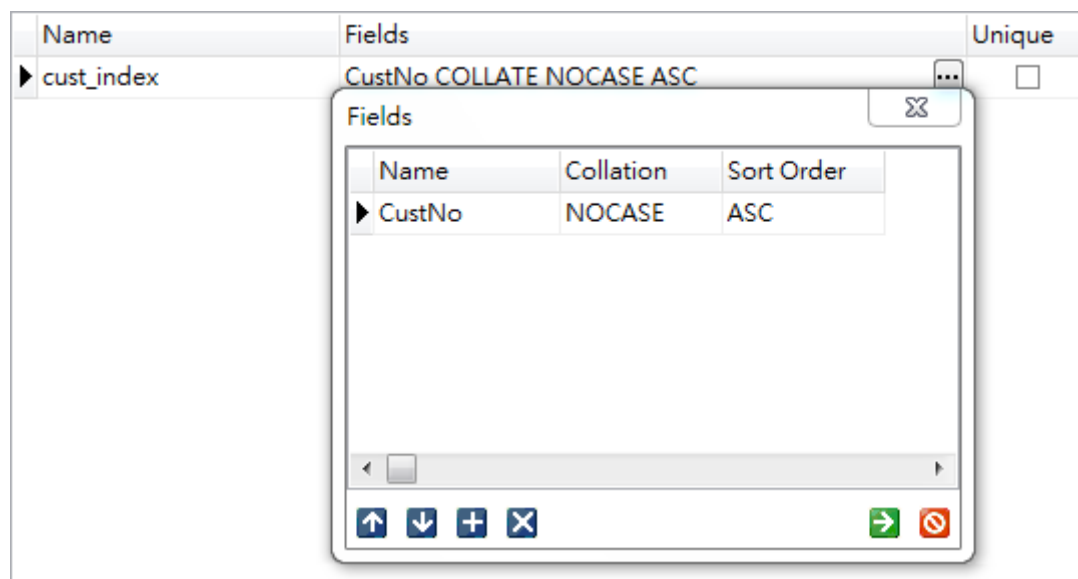
- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Just simply click/double-click on the index to edit.

### Delete Index

To delete a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Right-click on the index to delete and select the  **Delete Index** from the popup menu or click the  **Delete Index** from the toolbar.
- Confirm deleting in the dialog window.

## Setting SQLite Table Index Properties



Use the **Name** edit box to set the index name.

To include field(s) in the index, just simply double-click the **Fields**

Select the field(s) from the list. To remove the fields from the index, uncheck them in the same way. You can also use the arrow buttons to change the index field(s) order.

### Collation

To define a collating sequence used for text entries in that column. The default collating sequence is the collating sequence defined for that column.

#### **BINARY**

Compares string data using memcmp(), regardless of text encoding.

#### **NOCASE**

The same as binary, except the 26 upper case characters of ASCII are folded to their lower case equivalents before the comparison is performed. Note that only ASCII characters are case folded. SQLite does not attempt to do full UTF case folding due to the size of the tables required.

#### **RTRIM**

The same as binary, except that trailing space characters are ignored.

**Note:** Support in SQLite 3.

## **Sort Order**

To indicate sort order - ascending "ASC" or descending "DESC".

## ☒ **Unique**

All values of the indexed column(s) must only occur once.



## SQLite Table Foreign Keys

A foreign key is a field in a relational table that matches the primary key column of another table.

Foreign Keys are managed on the **Foreign Keys** tab of the Table Designer. Just simply click/double-click a foreign key field for editing. A right-click displays the popup menu or using the foreign key toolbar, allowing you to create new, edit and delete the selected foreign key field.

### Add Foreign Key

To add a foreign key

- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Right-click and select the  **Add Foreign Key** from the popup menu or click the  **Add Foreign Key** from the toolbar.
- Edit foreign key properties.



### Edit Foreign Key

To edit a foreign key


- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Just simply click/double-click on the foreign key to edit.

### Delete Foreign Key

To delete a foreign key


- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Right-click on the foreign key to delete and select the  **Delete Foreign Key** from the popup menu or click the  **Delete Foreign Key** from the toolbar.
- Confirm deleting in the dialog window.

## Setting SQLite Table Foreign Key Properties

Name	Fields	Reference Table	Reference Fields	On Delete	On Update
▶ cust_order_fk	CustNo	customer	CustNo 	NO ACTION	NO ACTION

Use the **Name** edit box to enter a name for the new key and then select a table field to include in the key from the **Fields** group.

Use the **Reference Table** dropdown list to select a foreign table respectively.

To include field(s) to the key, just simply double-click the **Fields/Reference Fields** field or click  to open the editor(s) for editing.

The **On Delete** and **On Update** dropdown list define the type of the actions to be taken.

### RESTRICT

The "RESTRICT" action means that the application is prohibited from deleting (for ON DELETE RESTRICT) or modifying (for ON UPDATE RESTRICT) a parent key when there exists one or more child keys mapped to it.

### NO ACTION

Configuring "NO ACTION" means just that: when a parent key is modified or deleted from the database, no special action is taken.

### CASCADE

A "CASCADE" action propagates the delete or update operation on the parent key to each dependent child key. For an "ON DELETE CASCADE" action, this means that each row in the child table that was associated with the deleted parent row is also deleted. For an "ON UPDATE CASCADE" action, it means that the values stored in each dependent child key are modified to match the new parent key values.

### SET NULL

If the configured action is "SET NULL", then when a parent key is deleted (for ON DELETE SET NULL) or modified (for ON UPDATE SET NULL), the child key columns of all rows in the child table that mapped to the parent key are set to contain SQL NULL values.

### SET DEFAULT

The "SET DEFAULT" actions are similar to "SET NULL", except that each of the child key columns is set to contain the columns default value instead of NULL.

## ☒ **Deferred**

Deferred foreign key constraints are not checked until the transaction tries to COMMIT.



## SQLite Table Uniques

Unique constraints ensure that the data contained in a column or a group of columns is unique with respect to all the rows in the table.

Uniques are managed on the **Uniques** tab of the Table Designer. Just simply click/double-click an unique field for editing. Using the unique toolbar, allowing you to create new, edit and delete the selected unique field.

### Add Unique

To add an unique

- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Right-click and select the  **Add Unique** from the popup menu or click the  **Add Unique** from the toolbar.
- Edit unique properties.



### Edit Unique

To edit an unique

- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Just simply click on the unique to edit.

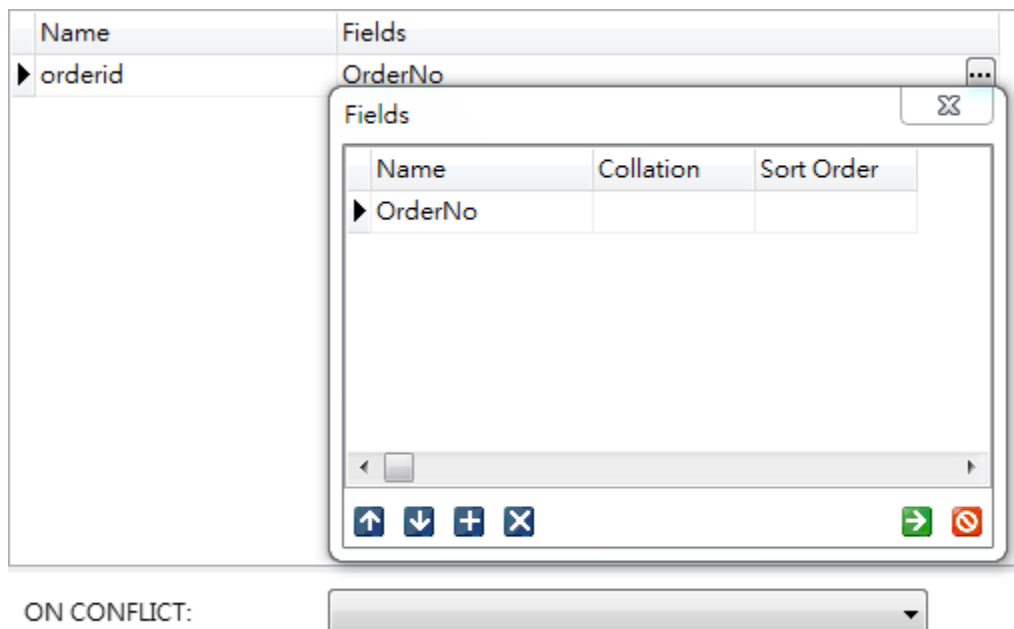
### Delete Unique

To delete an unique


- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Right-click on the unique to delete and select the  **Delete Unique** from the popup menu or click the  **Delete Unique** from the toolbar.
- Confirm deleting in the dialog window.

## Setting SQLite Table Unique Properties

Use the **Name** edit box to set the unique name.



### Fields

To set field(s) as unique, just simply double-click the **Fields** field or click  to open the editor(s) for editing.

### Collation

To define a collating sequence used for text entries in that column. The default collating sequence is the collating sequence defined for that column.

#### **BINARY**

Compares string data using memcmp(), regardless of text encoding.

#### **NOCASE**

The same as binary, except the 26 upper case characters of ASCII are folded to their lower case equivalents before the comparison is performed. Note that only ASCII characters are case folded. SQLite does not attempt to do full UTF case folding due to the size of the tables required.

#### **RTRIM**

The same as binary, except that trailing space characters are ignored.

**Note:** Support in SQLite 3.



## **Sort Order**

To indicate sort order - ascending "ASC" or descending "DESC".

## **ON CONFLICT**

To specify an algorithm used to resolve constraint conflicts. The default conflict resolution algorithm is ABORT.

### **ROLLBACK**

When a constraint violation occurs, an immediate ROLLBACK occurs, thus ending the current transaction, and the command aborts with a return code of SQLITE\_CONSTRAINT. If no transaction is active (other than the implied transaction that is created on every command) then this algorithm works the same as ABORT.

### **ABORT**

When a constraint violation occurs, the command backs out any prior changes it might have made and aborts with a return code of SQLITE\_CONSTRAINT. But no ROLLBACK is executed so changes from prior commands within the same transaction are preserved. This is the default behavior.

### **FAIL**

When a constraint violation occurs, the command aborts with a return code SQLITE\_CONSTRAINT. But any changes to the database that the command made prior to encountering the constraint violation are preserved and are not backed out. For example, if an UPDATE statement encountered a constraint violation on the 100th row that it attempts to update, then the first 99 row changes are preserved but changes to rows 100 and beyond never occur.

### **IGNORE**

When a constraint violation occurs, the one row that contains the constraint violation is not inserted or changed. But the command continues executing normally. Other rows before and after the row that contained the constraint violation continue to be inserted or updated normally. No error is returned when the IGNORE conflict resolution algorithm is used.

## **REPLACE**

When a UNIQUE constraint violation occurs, the pre-existing rows that are causing the constraint violation are removed prior to inserting or updating the current row. Thus the insert or update always occurs. The command continues executing normally following REPLACE. No error is returned by the REPLACE conflict resolution. If a NOT NULL constraint violation occurs, the NULL value is replaced by the default value for that column. If the column has no default value, then the ABORT algorithm is used. If a CHECK constraint violation occurs then the IGNORE algorithm is used.

## SQLite Table Checks



A check constraint allows you to specify that the value in a certain column must satisfy a Boolean (truth-value) expression.

**Note:** Checks are supported from SQLite version 3.3.0 or later.

Checks are managed on the **Checks** tab of the Table Designer. Just simply click/double-click a check field for editing. Using the check toolbar, allowing you to create new, edit and delete the selected check field.

### Add Check

To add a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Right-click and select the  **Add Check** from the popup menu or click the  **Add Check** from the toolbar.
- Edit check properties.



### Edit Check

To edit a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Just simply click on the check to edit.

### Delete Check

To delete a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Right-click on the check to delete and select the  **Delete Check** from the popup menu or click the  **Delete Check** from the toolbar.
- Confirm deleting in the dialog window.

## Setting SQLite Table Check Properties

Use the **Name** edit box to set the check name.

### Check

Set the condition for checking, e.g. "field\_name1 > 0 AND field\_name2 > field\_name1" in the **Check** edit box.

### Definition

Type in the definition for the check constraint.



## SQLite Table Triggers

A trigger is a database operation that is automatically performed when a specified database event occurs.

Triggers are managed on the **Triggers** tab of the Table Designer. Just simply click a trigger field for editing. A right-click displays the popup menu or using the trigger toolbar, allowing you to create new, edit and delete the selected trigger field.

### Add Trigger

To add a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Right-click and select the  **Add Trigger** from the popup menu or click the  **Add Trigger** from the toolbar.
- Edit trigger properties.



### Edit Trigger

To edit a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Just simply click on the trigger to edit.

### Delete Trigger

To delete a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Right-click on the trigger to delete and select the  **Delete Trigger** from the popup menu or click the  **Delete Trigger** from the toolbar.
- Confirm deleting in the dialog window.

## Setting SQLite Table Trigger Properties

### Name

Set the trigger name.

### Fires

Determine when the trigger actions will be executed relative to the insertion, modification or removal of the associated row.

### INSERT

Fires the trigger whenever an INSERT statement adds a row to a table.

### UPDATE

Fires the trigger whenever an UPDATE statement changes a value in one of the columns specified in **Update of Fields**. If no **Update of Fields** are present, the trigger will be fired whenever an UPDATE statement changes a value in any column of the table.

### DELETE

Fires the trigger whenever a DELETE statement removes a row from the table.

### Update of Fields

Specify the fields for UPDATE statement trigger upon necessary.

### Definition

Type in the definition for the trigger.

### Advanced

#### When Clause

Specify the trigger condition, which is a SQL condition that must be satisfied for the database to fire the trigger.

## SQLite Table Options

Table Options are managed on the **Options** tab of the Table Designer. Just simply click an option for editing.

### Primary Key ON CONFLICT

To specify an algorithm used to resolve primary key constraint conflicts. The default conflict resolution algorithm is ABORT.

#### ROLLBACK

When a constraint violation occurs, an immediate ROLLBACK occurs, thus ending the current transaction, and the command aborts with a return code of SQLITE\_CONSTRAINT. If no transaction is active (other than the implied transaction that is created on every command) then this algorithm works the same as ABORT.

#### ABORT

When a constraint violation occurs, the command backs out any prior changes it might have made and aborts with a return code of SQLITE\_CONSTRAINT. But no ROLLBACK is executed so changes from prior commands within the same transaction are preserved. This is the default behavior.

#### FAIL

When a constraint violation occurs, the command aborts with a return code SQLITE\_CONSTRAINT. But any changes to the database that the command made prior to encountering the constraint violation are preserved and are not backed out. For example, if an UPDATE statement encountered a constraint violation on the 100th row that it attempts to update, then the first 99 row changes are preserved but changes to rows 100 and beyond never occur.

#### IGNORE

When a constraint violation occurs, the one row that contains the constraint violation is not inserted or changed. But the command continues executing normally. Other rows before and after the row that contained the constraint violation continue to be inserted or updated normally. No error is returned when the IGNORE conflict resolution algorithm is used.

## **REPLACE**

When a UNIQUE constraint violation occurs, the pre-existing rows that are causing the constraint violation are removed prior to inserting or updating the current row. Thus the insert or update always occurs. The command continues executing normally following REPLACE. No error is returned by the REPLACE conflict resolution. If a NOT NULL constraint violation occurs, the NULL value is replaced by the default value for that column. If the column has no default value, then the ABORT algorithm is used. If a CHECK constraint violation occurs then the IGNORE algorithm is used.


## **Auto Increment**

Set/Reset the **Auto Increment** value in the edit field. The **Auto Increment Value** indicates the value for next record.





## SQLite Views


Views are useful for allowing users to access a set of tables as if it were a single table, and limiting their access to just that. Views can also be used to restrict access to rows (a subset of a particular table).

Just simply click  to open an object pane for **View**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit, open and delete the selected view.

### Create View

To create a new view

- Select anywhere on the object pane.
- Click the  **New View** from the object pane toolbar.  
or
- Right-click and select  **New View** from the popup menu.
- Edit view properties on the appropriate tabs of the View Designer.

**Hint:** To create new view you can also right-click the Views node of the navigation pane and select the  **New View** from the popup menu.

To create a new view with the same properties as one of the existing views has (using drag and drop method)




**Apply to:** current database {same connection}

- Select the view(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen view(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created view(s) will be named as "viewname\_**copy**".




**Apply to:** different database {same connection}  
different database {different connection} (Data Transfer tool will be activated)

- Select the view(s) for copying in the object pane.
- Drag and drop the chosen view(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel

To create a new view with modification as one of the existing views

- Select the view for modifying in the navigation pane/object pane.
- Right-click and select the  **Design View** from the popup menu.  
or
- Click the  **Design View** from the object pane toolbar.
- Modify view properties on the appropriate tabs of the View Designer.
- Click  **Save As**.

To create a new view with loading from a SQL file

- Select anywhere on the object pane.
- Click the  **New View** from the object pane toolbar.  
or
- Right-click and select  **New View** from the popup menu.
- Click  **Load**.

## Create View Shortcut

To create a view shortcut



- Select the view for editing in the navigation pane/object pane.
- Right-click and select **Create Open View Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.

**Note:** This option is used to provide a convenient way for you to open your view for entering data directly (Grid View/Form View) without activating the main Navicat.



## Edit View

To edit the existing view (manage its SQL definition etc)



- Select the view for editing in the navigation pane/object pane.
- Right-click and select the  **Design View** from the popup menu.  
or
- Click the  **Design View** from the object pane toolbar.
- Edit view properties on the appropriate tabs of the View Designer.

To change the name of the view

- Select the view for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.



## Open View

To open a view (manage view data)

- Select the view for opening in the navigation pane/object pane.
- Right-click and select the  **Open View** from the popup menu or simply double-click the view.  
or
- Click the  **Open View** from the object pane toolbar.

## Delete View

To delete a view

- Select the view for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete View** from the popup menu.  
or
- Click the  **Delete View** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve View Information

To achieve a view information

- Select the view in the navigation pane/object pane.
- Right-click the selected view and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQLite View Designer

**View Designer** is the basic Navicat tool for working with views. It allows you to create new view and edit the existing view definition (view name and the SELECT statement it implements).

- [Working with View Builder](#)
- [Editing View SQL Definition](#)
- View SQL Preview
- [View Preview](#)
- [View Explain](#)

## Working with SQLite View Builder (Available only in Full Version)

**View Builder** allows you to build views visually. It allows you to create and edit views without knowledge of SQL. See Query Builder for details.

## Editing SQLite View SQL Definition


The **Definition** tab allows you to edit the view definition as SQL statement (SELECT statement it implements).

Example:

```
SELECT
    clients.RecordID
FROM
    clients
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.


## SQLite View Preview

To preview the result of the view, click  **Preview** on the toolbar. If the query statement is correct, the **Result** and **Message** tabs will be opened.


The **Result** tab displays the data of the view as a grid and the **Message** tab displays the message log.



## SQLite View Explain

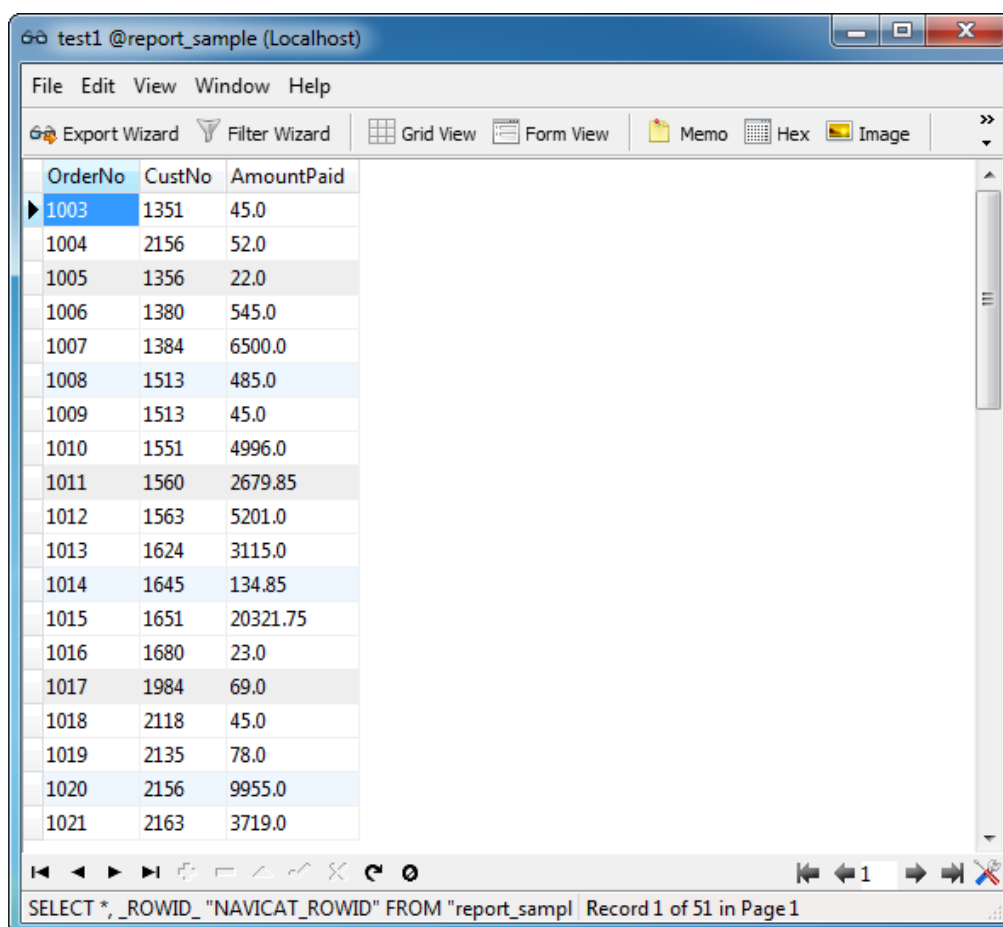
To return information about how the SQL statement would have operated, click  **Explain** on the toolbar. If the query statement is correct, the **Explain** tab will show.

## SQLite View Viewer

**View Viewer** displays the view data as a grid. Data can be displayed in three modes:  **Grid View**,  **Form View** and **Text/Blob View**. See Data View for details.


The toolbars of View Viewer provides the following functions for managing data:

- **Export Data**  
Export data to MS Word, MS Excel, MS Access, TXT, DBF, HTML, SQL, RTF and more.
- **Filter Data**  
Allow you to filter records by creating and applying filter criteria for the data grid.
- **Edit TEXT/BLOB**  
Allow you to view and edit the content of TEXT and BLOB fields.





## SQLite Indexes

Index provides a faster access path to table data. It is created using one or more columns of a table to speed SQL statement execution on that table.




Just simply click  to open an object pane for **Index**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected Index.

### Create Index

To create a new index



- Select anywhere on the object pane.
- Click the  **New Index** from the object pane toolbar.  
or
- Right-click and select  **New Index** from the popup menu.
- Edit index properties on the appropriate tabs of the Index Designer.

To create a new index with modification as one of the existing index

- Select the index for modifying in the object pane.
- Right-click and select the  **Design Index** from the popup menu or simply double-click the index.  
or
- Click the  **Design Index** from the object pane toolbar.
- Modify index properties on the appropriate tabs of the Index Designer.
- Click  **Save As**.

### Edit Index

To edit the existing index (manage its properties etc)

- Select the index for editing in the object pane.
- Right-click and select the  **Design Index** from the popup menu or simply double-click the index.  
or
- Click the  **Design Index** from the object pane toolbar.
- Edit index properties on the appropriate tabs of the Index Designer.

To change the name of the index

- Select the index for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.



## Maintain Index

To maintain an index

- Select the index for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - ReIndex

## Delete Index

To delete an index

- Select the index for deleting in the object pane.
- Right-click and select the  **Delete Index** from the popup menu.  
or
- Click the  **Delete Index** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Index Information

To achieve an index information (Index Owner, Index Type and DDL, etc)

- Select the index in the object pane.
- Right-click the selected index and choose **Index Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQLite Index Designer

**Index Designer** is the basic Navicat tool for working with indexes. It allows you to create new index and edit the existing index properties.

- [Editing Index General](#)
- Index SQL Preview

## Editing SQLite Index General

### Type

The types of the index.

#### **Normal**

A normal index does not impose restrictions on the column values.

#### **Unique**

An unique index indicates that no two rows of a table have duplicate values in the key columns.

### Table name

The table that contains the index.

### Fields

#### **Name**

To define the field.

#### **Collate**

To define a collating sequence used for text entries in that column. The default collating sequence is the collating sequence defined for that column.

#### **BINARY**

Compares string data using memcmp(), regardless of text encoding.

#### **NOCASE**

The same as binary, except the 26 upper case characters of ASCII are folded to their lower case equivalents before the comparison is performed. Note that only ASCII characters are case folded. SQLite does not attempt to do full UTF case folding due to the size of the tables required.

#### **RTRIM**

The same as binary, except that trailing space characters are ignored.

**Note:** Support in SQLite 3.


### Sort Order

To indicate sort order - ascending "ASC" or descending "DESC".

## SQLite Triggers



Triggers are database operations that are automatically performed when a specified database event occurs.

See [Triggers](#) for details.




Just simply click  to open an object pane for **Trigger**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected trigger.

### Create Trigger

To create a new trigger



- Select anywhere on the object pane.
- Click the  **New Trigger** from the object pane toolbar.  
or
- Right-click and select  **New Trigger** from the popup menu.
- Edit trigger properties on the appropriate tabs of the Trigger Designer.

To create a new trigger with modification as one of the existing trigger

- Select the trigger for modifying in the object pane.
- Right-click and select the  **Design Trigger** from the popup menu or simply double-click the trigger.  
or
- Click the  **Design Trigger** from the object pane toolbar.
- Modify trigger properties on the appropriate tabs of the Trigger Designer.
- Click  **Save As**.

## Edit Trigger

To edit the existing trigger (manage its general, advance, etc)



- Select the trigger for editing in the object pane.
- Right-click and select the  **Design Trigger** from the popup menu or simply double-click the trigger.  
or
- Click the  **Design Trigger** from the object pane toolbar.
- Edit trigger properties on the appropriate tabs of the Trigger Designer.

To change the name of the trigger

- Select the trigger for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

## Delete Trigger

To delete a trigger

- Select the trigger for deleting in the object pane.
- Right-click and select the  **Delete Trigger** from the popup menu.  
or
- Click the  **Delete Trigger** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Trigger Information

To achieve a trigger information (Trigger Owner, DB Link and DDL, etc)

- Select the trigger in the object pane.
- Right-click the selected trigger and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.



## SQLite Trigger Designer

**Trigger Designer** is the basic Navicat tool for working with triggers. It allows you to create new trigger and edit the existing trigger definition.

- [Editing Trigger General](#)
- [Editing Trigger Definition](#)
- Trigger SQL Preview

## Editing SQLite Trigger General

### Trigger Type

Define the trigger type: TABLE or VIEW.

### Table name or View name

Choose a table or view.

### Fires

Determine when the trigger actions will be executed relative to the insertion, modification or removal of the associated row.

### When

Specify the trigger condition for the database to fire the trigger.

### On Event

It indicates the kind of statement that activates the trigger.

#### ☒ Insert

The trigger is activated whenever adding a row to a table.

#### ☒ Delete

The trigger is activated whenever removing a row from the table.

#### ☒ Update

The trigger is activated whenever changing a value in one of the fields selected in **Update Of Fields**.

### Update of Fields

Specify the fields for UPDATE statement trigger upon necessary.

## Editing SQLite Trigger Definition

The **Definition** tab allows you to edit valid SQL statements in the trigger definition inside *BEGIN* and *END*.

## SQL Server Database Object Management

The following list contains the most common SQL Server database objects supported by Navicat.

- [Schemas](#)
- [Tables](#)
- [Views](#)
- [Functions/Procedures](#)
- [Indexes](#)
- [Synonyms](#)
- [Triggers](#)
- [Linked Servers](#)
- [Server Triggers](#)
- [Assemblies](#)
- [Database Triggers](#)
- [Partition Functions](#)
- [Partition Schemes](#)

## SQL Server Schemas



A schema contains named objects (tables, views, functions, etc) whose names may duplicate those of other objects existing in other schemas.

The schema name must be distinct from any existing schema name in the current database.

### Create Schema

**Note:** Support from SQL Server 2005 or later and SQL Azure.


To create a new schema

- Right-click the database in the navigation pane and choose  **New Schema....**  
or
- Right-click any existing schema and choose  **New Schema....**
- Edit schema properties on the appropriate tabs of the Schema Designer.

### Edit Schema


**Note:** Support from SQL Server 2005 or later and SQL Azure.

To edit the existing schema(manage its general etc)

- Right-click the schema in the navigation pane and choose  **Schema Properties....**
- Edit schema properties on the appropriate tabs of the Schema Designer.


### Open Schema

To open a schema which shows in the navigation pane

- Double-click the schema to open in the navigation pane.  
or
- Right-click the schema and choose  **Open Schema.**

### Close Schema


To close a schema

- Right-click the schema in the navigation pane and choose  **Close Schema.**

## Delete Schema

**Note:** Support from SQL Server 2005 or later and SQL Azure.

To delete a schema

- Right-click the schema in the navigation pane and choose  **Delete Schema**.
- Confirm deleting in the dialog window.

## SQL Server Schema Designer

**Schema Designer** is the basic Navicat tool for working with schema. It allows you to create new schema and edit the existing schema properties.

**Note:** Support from SQL Server 2005 or later and SQL Azure.

- [Editing Schema General](#)
- Editing Schema Comment (SQL Azure does not support)
- Schema SQL Preview

## Editing SQL Server Schema General

### Schema Name

The name of a schema which is identified within the database.

### Owner


The name of the database-level principal that will own the schema. This principal may own other schemas, and may not use the current schema as its default schema.







## SQL Server Tables


Tables are database objects that contain all the data in a database. A table definition is a collection of columns. In tables, data is organized in a row-and-column format similar to a spreadsheet. Each row represents a unique record, and each column represents a field within the record.

Just simply click  to open an object pane for **Table**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit, open and delete the selected table.

### Create Table

To create a new table

- Select anywhere on the object pane.
- Click the  **New Table** from the object pane toolbar.  
or
- Right-click and select  **New Table** from the popup menu.
- Edit table properties and fields on the appropriate tabs of the Table Designer.

**Hint:** To create new table you can also right-click the Tables node of the navigation pane and select the  **New Table** from the popup menu.

To create a new table with the same properties as one of the existing tables has (using popup menu)

**Apply to:** current database {same connection}

- Select the table(s) for copying in the navigation pane/object pane.
- Right-click and select the **Duplicate Table** from the popup menu.
- The newly created table(s) will be named as "tablename\_**copy**".

To create a new table with the same properties as one of the existing tables has (using drag and drop method)

**Apply to:** current database {same connection}




- Select the table(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen table(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created table(s) will be named as "tablename\_**copy**"

**Apply to:** different database {same connection}

different database {different connection (same or cross server type)} (Data Transfer tool will be activated)

- Select the table(s) for copying in the object pane.
- Drag and drop the chosen table(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel

To create a new table with modification as one of the existing tables

- Select the table for modifying in the navigation pane/object pane.
- Right-click and select the  **Design Table** from the popup menu.  
or
- Click the  **Design Table** from the object pane toolbar.
- Modify table properties and fields on the appropriate tabs of the Table Designer.
- Click  **Save As**.

## Create Table Shortcut



To create a table shortcut

- Select the table for editing in the navigation pane/object pane.
- Right-click and select **Create Open Table Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.

**Note:** This option is used to provide a convenient way for you to open your table for entering data directly (Grid View/Form View) without activating the main Navicat.

## Edit Table

To edit the existing table (manage its fields, indexes, foreign keys and triggers etc)



- Select the table for editing in the navigation pane/object pane.
- Right-click and select the  **Design Table** from the popup menu.  
or
- Click the  **Design Table** from the object pane toolbar.
- Edit table properties and fields on the appropriate tabs of the Table Designer.


To change the name of the table

- Select the table for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.

## Open Table (manage table data)

To open a table

- Select the table for opening in the navigation pane/object pane.
- Right-click and select the  **Open Table** from the popup menu or simply double-click the table.  
or
- Click the  **Open Table** from the object pane toolbar.

**Note:** This option is only applied if you do wish Navicat loads all your images while opening the table. To open the graphical table with faster performance, use  **Open Table (Quick)** below.

To open a table with graphical fields

- Select the table for opening in the navigation pane/object pane.
- Right-click and select the  **Open Table (Quick)** from the popup menu.

**Note:** Faster performance for opening the graphical table, as BLOB fields (images) will not be loaded until you click on the cell.



## Empty Table

To empty a table

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Empty Table** from the popup menu.

## Delete Table

To delete a table

- Select the table for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Table** from the popup menu.  
or
- Click the  **Delete Table** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Table Information

To achieve a table information

- Select the table in the navigation pane/object pane.
- Right-click the selected table and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQL Server Table Designer

**Table Designer** is the basic Navicat tool for working with tables. It allows you to create, edit and drop table's fields, indexes, foreign keys, and much more.



- [Managing Table Fields](#)
- [Managing Table Indexes](#)
- [Managing Table Foreign Keys](#)
- [Managing Table Uniques](#)
- [Managing Table Checks](#)
- [Managing Table Triggers](#)
- [Managing Table Options](#)
- [Managing Table Storage](#)
- Managing Table Comment (SQL Azure does not support)
- Table SQL Preview

## SQL Server Table Fields

Table fields are managed on the **Fields** tab of the Table Designer. Just simply click a field for editing. A right-click displays the popup menu or using the field toolbar, allowing you to create new and drop the selected field.

### Add Field

To add a field to the table

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click and select the  **Add Field** from the popup menu or click the  **Add Field** from the toolbar.
- Edit field properties.

To add a new field with modification as one of the existing fields

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Select field.
- Right-click and select the **Duplicate Field** from the popup menu.
- Edit field properties.



### Edit Field

To edit the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Simply click on the field to edit.

## Delete Field

To delete the table field

- Open the table in the Table Designer.
- Open the **Fields** tab.
- Right-click on the field to delete and select the  **Delete Field** from the popup menu or click the  **Delete Field** from the toolbar.
- Confirm deleting in the dialog window.

## Setting SQL Server Table Field Properties

Name	Type	Length	Scale	Allow Null	
EmployeeID	int	0	0	<input type="checkbox"/>	 1
NationalIDNumber	nvarchar	15	0	<input type="checkbox"/>	
ContactID	int	0	0	<input type="checkbox"/>	
LoginID	nvarchar	256	0	<input type="checkbox"/>	
ManagerID	int	0	0	<input checked="" type="checkbox"/>	

### Name

The Name is a descriptive identifier for a field that can be up to 128 characters. The names should be descriptive enough that anyone can easily identify them when viewing or editing records. For example, LastName, FirstName, StreetAddress, or HomePhone.

Use the **Name** edit box to set the field name. Note that the name of the field must be unique among all the field names in the table.

### Type

After you name a field, you choose a data type for the data to be contained in the field. When you choose a field's data type, you are deciding:

- What kind of values to allow in the field. You cannot store text in field with the **Numeric** data type.
- How much storage space SQL Server is to set aside for the data in that field.
- What types of operations can be performed on the values in that field.

The **Type** dropdown list defines the type of the field data. See [SQL Server Data Type](#) and [SQL Azure Support Data Type](#) for details.

### Length and Scale

Use the **Length** edit box to define the length of the field and use **Scale** edit box to define the number of digits after the decimal point (the scale) for Floating Point data type.

**Note:** Be careful when shortening the field length as losing data might be caused.

### ☒ Allow Null

Allow the NULL values for the field.



## **Primary Key**

A **Primary Key** is a single field or combination of fields that uniquely defines a record. None of the fields that are part of the primary key can contain a null value.

### **Primary Key Name**

Right-click and select **Primary Key Name** from the popup menu to enter the primary key constraint name.

### **Fill Factor**

Right-click and select **Fill Factor** from the popup menu to specify how full the Database Engine should make each index page that is used to store the index data. User-specified fillfactor values can be from 1 through 100.

### **Design Primary Key Index**

Right-click and select **Design Primary Key Index** from the popup menu to design primary key index.

## Setting Other SQL Server Table Field Properties

For **bigint**, **decimal**, **int**, **numeric**, **smallint**, **tinyint** data types:

### ☒ **Identity**

Indicate that the new column is an identity column.

For **uniqueidentifier** data types:

### ☒ **Row GUID**

Indicate that the new column is a row GUID column. Only one uniqueidentifier column per table can be designated as the ROWGUIDCOL column.

**Note:** SQL Azure does not support.

For **char**, **nchar**, **ntext**, **nvarchar**, **text**, **varchar** data types:

### **Collation**

Specify the collation for the column.

For **xml** data types:

### ☒ **Column Set For All Sparse Columns**

Combine all of the sparse columns of a table into a structured output.

**Note:** Support from SQL Server 2008 or later.

For **varbinary(MAX)** data types:

### ☒ **File Stream**

Specify FILESTREAM storage for the varbinary(max) BLOB data.

**Note:** Support from SQL Server 2008 or later.

For **User Defined Type** data types:

### **User Defined Type Schema**

Set the schema of the user defined type.

### **User Defined Type**

Set the user defined type.

For **Computed Column** data types:

## **Computed Expression**

Set an expression that defines the value of a computed column.

### ☒ **Persisted**

Specify that the SQL Server Database Engine will physically store the computed values in the table, and update the values when any other columns on which the computed column depends are updated.

For most data types:

## **Default**

Set the default value for the field.

### ☒ **With Values**

Check this to enable the with values option.

### ☒ **Sparse**

Indicate that the column is a sparse column.

**Note:** Support from SQL Server 2008 or later.

For all data types:

## **Comment**

Set any optional text describing the current field.

**Note:** SQL Azure does not support.



## SQL Server Table Indexes

Indexes are optional structures associated with tables. You can create indexes on one or more columns of a table to speed SQL statement execution on that table.

Table indexes are managed on the **Indexes** tab of the Table Designer. Just simply click/double-click an index field for editing. A right-click displays the popup menu or using the index toolbar, allowing you to create new, edit and delete the selected index field.

### Add Index

To add a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Right-click and select the  **Add Index** from the popup menu or click the  **Add Index** from the toolbar.
- Edit index properties.



### Edit Index

To edit a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Just simply click/double-click on the index to edit.

### Delete Index


To delete a table index

- Open the table in the Table Designer.
- Open the **Indexes** tab.
- Right-click on the index to delete and select the  **Delete Index** from the popup menu or click the  **Delete Index** from the toolbar.
- Confirm deleting in the dialog window.

## Setting SQL Server Table Index Properties

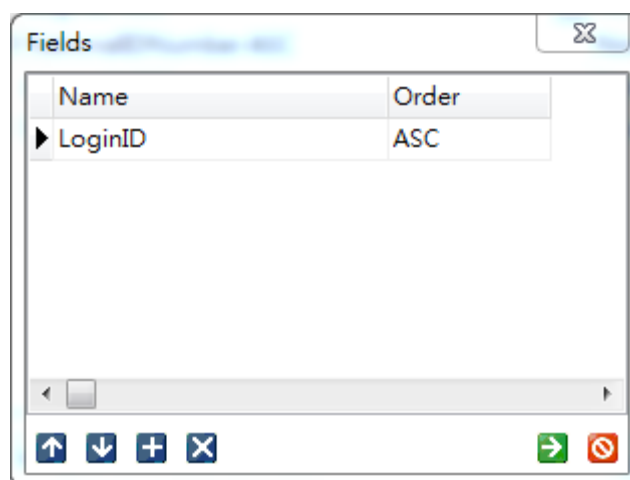
Name	Fields	Index Type	Unique
▶ AK_Employee_LoginID	LoginID ASC	Non-Clustered	<input checked="" type="checkbox"/>
AK_Employee_NationalID	NationalIDNumber ASC	Non-Clustered	<input checked="" type="checkbox"/>

Use the **Name** edit box to set the index name.

To include field(s) in the index, just simply double-click the **Fields** field or click  to open the editor for editing.

Select the field(s) from the list. To remove the fields from the index, uncheck them in the same way. You can also use the arrow buttons to change the index field(s) order. The **Order** dropdown list is used to set sort direction.

**Note:** Some of data types do not allow indexing. For example: text



The **Index Type** dropdown list defines the type of the table index.

### Clustered

Create an index in which the logical order of the key values determines the physical order of the corresponding rows in a table.

### Non-Clustered

Create an index that specifies the logical ordering of a table. With a nonclustered index, the physical order of the data rows is independent of their indexed order.

## **Spatial**

Create a spatial index on a specified table and column. An index can be created before there is data in the table.

**Note:** Support from SQL Server 2008 or later and SQL Azure.

## **XML**

Create an XML index on a specified table. An index can be created before there is data in the table.

**Note:** Support from SQL Server 2005 or later.

## **Unique**

Create a unique index on a table.

## **Comment**

Specify the comment of the index.

**Note:** SQL Azure does not support.



## SQL Server Table Foreign Keys

A foreign key is a field in a relational table that matches the primary key column of another table. The foreign key can be used to cross-reference tables.

Foreign Keys are managed on the **Foreign Keys** tab of the Table Designer. Just simply click/double-click a foreign key field for editing. A right-click displays the popup menu or using the foreign key toolbar, allowing you to create new, edit and delete the selected foreign key field.

### Add Foreign Key

To add a foreign key

- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Right-click and select the **Add Foreign Key** from the popup menu or click the **Add Foreign Key** from the toolbar.
- Edit foreign key properties.



### Edit Foreign Key

To edit a foreign key

- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Just simply click/double-click on the foreign key to edit.

### Delete Foreign Key

To delete a foreign key


- Open the table in the Table Designer.
- Open the **Foreign Keys** tab.
- Right-click on the foreign key to delete and select the **Delete Foreign Key** from the popup menu or click the **Delete Foreign Key** from the toolbar.
- Confirm deleting in the dialog window.

## Setting SQL Server Table Foreign Key Properties

Name	Fields	Reference Schema	Reference Table	Reference Fields	On Delete	On Update	Enable	Not For Replication
FK_Employee_Cor	ContactID	Person	Contact	ContactID	NO ACTION	NO ACTION	<input checked="" type="checkbox"/>	
FK_Employee_Emp	ManagerID	HumanResources	Employee	EmployeeID	NO ACTION	NO ACTION	<input checked="" type="checkbox"/>	

Use the **Name** edit box to enter a name for the new key and then select a table field to include in the key from the **Fields** group.

Use the **Reference Schema** and **Reference Table** dropdown lists to select a foreign schema and table respectively.

To include field(s) to the key, just simply double-click the **Fields/Reference Fields** field or click  to open the editor(s) for editing.

The **On Delete** and **On Update** dropdown list define the type of the actions to be taken.

### No Action

The Database Engine raises an error and the delete or update action on the row in the parent table is rolled back.

### Cascade

Corresponding rows are deleted from or updated in the referencing table if that row is deleted from or updated in the parent table.

### Set Null

All the values that make up the foreign key are set to NULL when the corresponding row in the parent table is deleted or updated.

### Set Default

All the values that make up the foreign key are set to their default values when the corresponding row in the parent table is deleted or updated.

### ☒ Enable

You can choose whether to enable / disable the foreign key constraint by checking / unchecking the box.



## ☒ **Not For Replication**

The constraint is not enforced when replication agents perform insert, update, or delete operations.

**Note:** SQL Azure does not support.

## **Comment**

Specify the comment of the foreign key.

**Note:** SQL Azure does not support.



## SQL Server Table Uniques

Unique constraints ensure that the data contained in a column or a group of columns is unique with respect to all the rows in the table.

Uniques are managed on the **Uniques** tab of the Table Designer. Just simply click/double-click an unique field for editing. Using the unique toolbar, allowing you to create new, edit and delete the selected unique field.

### Add Unique

To add an unique

- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Right-click and select the  **Add Unique** from the popup menu or click the  **Add Unique** from the toolbar.
- Edit unique properties.



### Edit Unique

To edit an unique

- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Just simply click on the unique to edit.

### Delete Unique

To delete an unique


- Open the table in the Table Designer.
- Open the **Uniques** tab.
- Right-click on the unique to delete and select the  **Delete Unique** from the popup menu or click the  **Delete Unique** from the toolbar.
- Confirm deleting in the dialog window.

## Setting SQL Server Table Unique Properties

Name	Fields	Clustered
▶ emp_uni	EmployeeID, LoginID	<input type="checkbox"/>

Use the **Name** edit box to set the unique name.

### Fields

To set field(s) as unique, just simply double-click the **Fields** field or click  to open the editor(s) for editing.

### ☒ Clustered

Indicate that a clustered index is created for the unique constraint.

### Comment

Specify the comment of the unique.

**Note:** SQL Azure does not support.



## SQL Server Table Checks

A check is a constraint that enforces domain integrity by limiting the possible values that can be entered into a column or columns.

Checks are managed on the **Checks** tab of the Table Designer. Just simply click/double-click a check field for editing. Using the check toolbar, allowing you to create new, edit and delete the selected check field.

### Add Check

To add a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Right-click and select the  **Add Check** from the popup menu or click the  **Add Check** from the toolbar.
- Edit check properties.



### Edit Check

To edit a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Just simply click on the check to edit.

### Delete Check

To delete a check

- Open the table in the Table Designer.
- Open the **Checks** tab.
- Right-click on the check to delete and select the  **Delete Check** from the popup menu or click the  **Delete Check** from the toolbar.
- Confirm deleting in the dialog window.

## Setting SQL Server Table Check Properties

Use the **Name** edit box to set the check name.

### Check

Set the logical expression that returns TRUE or FALSE, e.g. "field\_name1 > 0 AND field\_name2 > field\_name1" in the **Check** edit box.

#### ☒ **Enable**

You can choose whether to enable / disable the check constraint by checking / unchecking the box.

#### ☒ **Not For Replication**

The constraint is not enforced when replication agents perform insert, update, or delete operations.

**Note:** SQL Azure does not support.

### Definition

Type in the definition for the check constraint.

### Comment

Specify the comment of the check.

**Note:** SQL Azure does not support.



## SQL Server Table Triggers

A trigger is a special kind of stored procedure that automatically executes when an event occurs in the database server.

Triggers are managed on the **Triggers** tab of the Table Designer. Just simply click a trigger field for editing. A right-click displays the popup menu or using the trigger toolbar, allowing you to create new, edit and delete the selected trigger field.

### Add Trigger

To add a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Right-click and select the  **Add Trigger** from the popup menu or click the  **Add Trigger** from the toolbar.
- Edit trigger properties.



### Edit Trigger

To edit a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Just simply click on the trigger to edit.

### Delete Trigger

To delete a trigger

- Open the table in the Table Designer.
- Open the **Triggers** tab.
- Right-click on the trigger to delete and select the  **Delete Trigger** from the popup menu or click the  **Delete Trigger** from the toolbar.
- Confirm deleting in the dialog window.

## Setting SQL Server Table Trigger Properties

Use the **Name** edit box to set the trigger name.

Use the **Fires** dropdown list to define the trigger action time.

### **AFTER**

Specify that the DML trigger is fired only when all operations specified in the triggering SQL statement have executed successfully.

### **INSTEAD OF**

Specify that the DML trigger is executed instead of the triggering SQL statement, therefore, overriding the actions of the triggering statements.

#### ☒ **Insert**

The trigger is activated whenever a new row is inserted into the table.

#### ☒ **Update**

The trigger is activated whenever a row is modified.

#### ☒ **Delete**

The trigger is activated whenever a row is deleted from the table.

#### ☒ **Enable**

You can choose whether to enable / disable the trigger constraint by checking / unchecking the box.

The **Definition** tab defines the statement to execute when the trigger activates. To include your statement, just simply click to write. If you want to execute multiple statements, use the **BEGIN ... END** compound statement construct.

## Advanced

### **Execute As**

Specify the security context under which the trigger is executed.

**Note:** Support from SQL Server 2005 or later and SQL Azure.

## User

Choose a user that the trigger executes in.

**Note:** Support from SQL Server 2005 or later and SQL Azure.

## ☒ Encrypted

Obfuscate the text of the CREATE TRIGGER statement.

**Note:** Support from SQL Server 2005 or later.

## ☒ Not For Replication

Indicate that the trigger should not be executed when a replication agent modifies the table that is involved in the trigger.

**Note:** SQL Azure does not support.

## ☒ With Append

Specify that an additional trigger of an existing type should be added.

**Note:** SQL Azure does not support.

## Definition Type

Choose the type of definition.

**Note:** Support from SQL Server 2005 or later.

## Comment

Specify the comment of the trigger.

**Note:** SQL Azure does not support.



## SQL Server Table Options

### Table Lock Escalation

Specify the allowed methods of lock escalation for a table.

**Note:** Support from SQL Server 2008 or later.

### Identity Seed

The value used for the very first row loaded into the table.

### Identity Increment

The incremental value added to the identity value of the previous row loaded.

### Current Identity Value

Set the current identity value.

**Note:** SQL Azure does not support.

### ☒ Not For Replication

Values are not incremented in identity columns when replication agents perform inserts.

**Note:** SQL Azure does not support.

### ☒ Change Tracking Enabled

Specify change tracking is enabled for the table.

**Note:** Support from SQL Server 2008 or later.

### ☒ Track Columns Updated

Specify the Database Engine tracks which change tracked columns were updated.

**Note:** Support from SQL Server 2008 or later.

## SQL Server Table Storage

SQL Azure does not support this tab.

### On Filegroup

#### Filegroup

Choose a filegroup that storing the table.

#### Text/Image Filegroup

Choose a filegroup for storing text, ntext, image, xml, varchar(max), nvarchar(max), varbinary(max), and CLR user-defined type columns.

#### File Stream Filegroup

Choose a filegroup for FILESTREAM data.

**Note:** Support from SQL Server 2008 or later.

### On Partition Scheme

**Note:** Support from SQL Server 2005 or later.

#### Partition Scheme

Choose a partition scheme that storing the table.

#### Partition Column

Choose a partition column name.

#### File Stream Partition Scheme

Choose a partition scheme for FILESTREAM data.

**Note:** Support from SQL Server 2008 or later.

### Data Compression

**Note:** Support from SQL Server 2008 or later.

#### Partition Number

The partition which the DATA\_COMPRESSION setting applies.

## Type

### **NONE**

Table or specified partitions are not compressed.

### **ROW**


Table or specified partitions are compressed by using row compression.

### **PAGE**

Table or specified partitions are compressed by using page compression.



## SQL Server Views


A view can be thought of as either a virtual table or a stored query. Unless a view is indexed, its data is not stored in the database as a distinct object. What is stored in the database is a SELECT statement. The result set of the SELECT statement forms the virtual table returned by the view. A user can use this virtual table by referencing the view name in Transact-SQL statements the same way a table is referenced.

Just simply click  to open an object pane for **View**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit, open and delete the selected view.

### Create View

To create a new view

- Select anywhere on the object pane.
- Click the  **New View** from the object pane toolbar.  
or
- Right-click and select  **New View** from the popup menu.
- Edit view properties on the appropriate tabs of the View Designer.

**Hint:** To create new view you can also right-click the Views node of the navigation pane and select the  **New View** from the popup menu.

To create a new view with the same properties as one of the existing views has (using drag and drop method)




**Apply to:** current database {same connection}

- Select the view(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen view(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created view(s) will be named as "viewname\_**copy**".




**Apply to:** different database {same connection}  
different database {different connection} (Data Transfer tool will be activated)

- Select the view(s) for copying in the object pane.
- Drag and drop the chosen view(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel

To create a new view with modification as one of the existing views

- Select the view for modifying in the navigation pane/object pane.
- Right-click and select the  **Design View** from the popup menu.  
or
- Click the  **Design View** from the object pane toolbar.
- Modify view properties on the appropriate tabs of the View Designer.
- Click  **Save As**.

To create a new view with loading from a SQL file

- Select anywhere on the object pane.
- Click the  **New View** from the object pane toolbar.  
or
- Right-click and select  **New View** from the popup menu.
- Click  **Load**.

## Create View Shortcut



To create a view shortcut

- Select the view for editing in the navigation pane/object pane.
- Right-click and select **Create Open View Shortcut...** from the popup menu.
- Define the location you wish your shortcut to be saved.

**Note:** This option is used to provide a convenient way for you to open your view for entering data directly (Grid View/Form View) without activating the main Navicat.

## Edit View

To edit the existing view (manage its SQL definition etc)



- Select the view for editing in the navigation pane/object pane.
- Right-click and select the  **Design View** from the popup menu.  
or
- Click the  **Design View** from the object pane toolbar.
- Edit view properties on the appropriate tabs of the View Designer.

To change the name of the view

- Select the view for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.



## Open View

To open a view (manage view data)

- Select the view for opening in the navigation pane/object pane.
- Right-click and select the  **Open View** from the popup menu or simply double-click the view.  
or
- Click the  **Open View** from the object pane toolbar.

## Delete View

To delete a view

- Select the view for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete View** from the popup menu.  
or
- Click the  **Delete View** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve View Information

To achieve a view information

- Select the view in the navigation pane/object pane.
- Right-click the selected view and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQL Server View Designer

**View Designer** is the basic Navicat tool for working with views. It allows you to create new view and edit the existing view definition (view name and the SELECT statement it implements).

- [Working with View Builder](#)
- [Editing View SQL Definition](#)
- [Setting Advanced View Properties](#)
- Editing View Comment (SQL Azure does not support)
- View SQL Preview
- [View Preview](#)
- [View Explain](#)



## **Working with SQL Server View Builder (Available only in Full Version)**

**View Builder** allows you to build views visually. It allows you to create and edit views without knowledge of SQL. See Query Builder for details.

## Editing SQL Server View SQL Definition

The **Definition** tab allows you to edit the view definition as SQL statement (SELECT statement it implements).

Example:

```
SELECT
    report_sample.clients.RecordID
FROM
    report_sample.clients
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see [Editor View and More Features](#).

## Setting Advanced SQL Server View Properties

### ☒ **Encrypted**

Encrypt the entries in sys.syscomments that contain the text of the CREATE VIEW statement.

**Note:** SQL Azure does not support.

### ☒ **Schema Bound**

Bind the view to the schema of the underlying table or tables.


### ☒ **View Metadata**

Specify that the instance of SQL Server will return to the DB-Library, ODBC, and OLE DB APIs the metadata information about the view, instead of the base table or tables, when browse-mode metadata is being requested for a query that references the view.

### ☒ **With Check Option**


Force all data modification statements executed against the view to follow the criteria set within select\_statement.

## SQL Server View Preview

To preview the result of the view, click  **Preview** on the toolbar. If the query statement is correct, the **Result** and **Message** tabs will be opened.

The **Result** tab displays the data of the view as a grid and the **Message** tab displays the message log.

## SQL Server View Explain

To return information about how the SQL statement would have operated, click  **Explain** on the toolbar. If the query statement is correct, the **Explain** tab will show.

The **Explain** tab displays the information as a grid:

Column	Description
StmtText	For rows that are not of type PLAN_ROW, this column contains the text of the Transact-SQL statement. For rows of type PLAN_ROW, this column contains a description of the operation. This column contains the physical operator and may optionally also contain the logical operator. This column may also be followed by a description that is determined by the physical operator.
StmtId	Number of the statement in the current batch.
NodeId	ID of the node in the current query.
Parent	Node ID of the parent step.
PhysicalOp	Physical implementation algorithm for the node. For rows of type PLAN_ROWS only.
LogicalOp	Relational algebraic operator this node represents. For rows of type PLAN_ROWS only.
Argument	Provides supplemental information about the operation being performed. The contents of this column depend on the physical operator.
DefinedValues	Contains a comma-separated list of values introduced by this operator. These values may be computed expressions which were present in the current query (for example, in the SELECT list or WHERE clause), or internal values introduced by the query processor in order to process this query. These defined values may then be referenced elsewhere within this query. For rows of type PLAN_ROWS only.
EstimateRows	Estimated number of rows of output produced by this operator. For rows of type PLAN_ROWS only.
EstimateIO	Estimated I/O cost* for this operator. For rows of type PLAN_ROWS only.

EstimateCPU	Estimated CPU cost* for this operator. For rows of type PLAN_ROWS only.
AvgRowSize	Estimated average row size (in bytes) of the row being passed through this operator.
TotalSubtreeCost	Estimated (cumulative) cost* of this operation and all child operations.
OutputList	Contains a comma-separated list of columns being projected by the current operation.
Warnings	Contains a comma-separated list of warning messages relating to the current operation. Warning messages may include the string "NO STATS:()" with a list of columns. This warning message means that the query optimizer attempted to make a decision based on the statistics for this column, but none were available. Consequently, the query optimizer had to make a guess, which may have resulted in the selection of an inefficient query plan.
Type	Node type. For the parent node of each query, this is the Transact-SQL statement type (for example, SELECT, INSERT, EXECUTE, and so on). For subnodes representing execution plans, the type is PLAN_ROW.
Parallel	0 = Operator is not running in parallel. 1 = Operator is running in parallel.
EstimateExecutions	Estimated number of times this operator will be executed while running the current query.

\* Cost units are based on an internal measurement of time, not wall-clock time. They are used for determining the relative cost of a plan in comparison to other plans.

## SQL Server View Viewer

**View Viewer** displays the view data as a grid. Data can be displayed in three modes:  **Grid View**,  **Form View** and **Text/Blob View**. See Data View for details.

The toolbars of View Viewer provides the following functions for managing data:

- **Commit**

Make permanent all changes performed in the transaction.

**Hint:** The Commit button is visible only when **Auto Commit** is disabled under Option Settings.

- **Rollback**

Undo work done in the current transaction.

**Hint:** The Rollback button is visible only when **Auto Commit** is disabled under Option Settings.

- **Export Data**

Export data to MS Word, MS Excel, MS Access, TXT, DBF, HTML, SQL, RTF and more.

- **Filter Data**

Allow you to filter records by creating and applying filter criteria for the data grid.

- **Edit TEXT/BLOB**

Allow you to view and edit the content of TEXT and BLOB fields.

vEmployeeDepartment @AdventureWorks.HumanResources (SQL Server Auth Connectio...

File Edit View Window Help

Commit Rollback Export Wizard Filter Wizard Grid View Form View

EmployeeID	Title	FirstName	MiddleName	LastName	Suffix	JobTitle
206	(Null)	Brian	P	LaMee	(Null)	Schedulin
207	(Null)	Kitti	H	Lertpiriyasuwat	(Null)	Productio
208	(Null)	Jay	G	Adams	(Null)	Productio
209	(Null)	Jan	S	Mikovsky	(Null)	Productio
210	(Null)	Brenda	M	Diaz	(Null)	Productio
211	(Null)	Andrew	M	Cencini	(Null)	Productio
212	(Null)	Chris	K	Norred	(Null)	Control S
213	(Null)	Chris	O	Okelberry	(Null)	Productio
214	(Null)	Shelley	N	Dyck	(Null)	Productio
215	(Null)	Gabe	B	Mares	(Null)	Productio
216	(Null)	Mike	K	Seamans	(Null)	Accounta
217	(Null)	Michael	(Null)	Raheem	(Null)	Research
218	(Null)	Gary	E.	Altman	III	Facilities
219	(Null)	Charles	B	Fitzgerald	(Null)	Productio
220	(Null)	Ebru	N	Ersan	(Null)	Productio
221	(Null)	Sylvester	A	Valdez	(Null)	Productio
222	(Null)	Brian	Richard	Goldstein	(Null)	Productio
223	(Null)	Linda	P	Meisner	(Null)	Buyer
224	(Null)	Betsy	A	Stadick	(Null)	Productio
225	(Null)	Magnus	E	Hedlund	(Null)	Facilities
226	(Null)	Karan	R	Khanna	(Null)	Productio
227	(Null)	Mary	R	Baker	(Null)	Productio
228	(Null)	Kevin	M	Homer	(Null)	Productio
229	(Null)	Mihail	U	Frintu	(Null)	Productio

SELECT \* FROM [HumanResources].[vEmployeeDepartment] Record 1 of 290 in page 1




## SQL Server Functions/Procedures

A user-defined function, which is a Transact-SQL or common language runtime (CLR) routine that accepts parameters, performs an action, such as a complex calculation, and returns the result of that action as a value. The return value can either be a scalar (single) value or a table.



Stored procedures are similar to procedures in other programming languages in that they can:

- Accept input parameters and return multiple values in the form of output parameters to the calling procedure or batch.
- Contain programming statements that perform operations in the database, including calling other procedures.
- Return a status value to a calling procedure or batch to indicate success or failure (and the reason for failure).

Just simply click  to open an object pane for **Function**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected function/procedure.

### Create Function/Procedure

To create a new function/procedure

- Select anywhere on the object pane.
- Click the  **New Function** from the object pane toolbar.  
or
- Right-click and select  **New Function** from the popup menu.
- Edit function/procedure properties on the appropriate tabs of the Function/Procedure Designer.

**Hint:** To create new function/procedure you can also right-click the Function node of the navigation pane and select the  **New Function** from the popup menu.

To create a new function/procedure with the same properties as one of the existing function/procedure has (using drag and drop method)

**Apply to:** current schema {same connection}



- Select the function/procedure(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen function/procedure(s) to the target location.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Move here
  - Cancel
- The newly created function/procedure(s) will be named as "function/procedurename\_**copy**".

**Apply to:** different schema {same connection}  
different schema {different connection} (Data Transfer tool will be activated)

- Select the function/procedure(s) for copying in the object pane.
- Drag and drop the chosen function/procedure(s) to the target database.
- Select one of the following options:
  - Copy here (Structure and Data)
  - Copy here (Structure only)
  - Cancel



## Edit Function/Procedure

To edit the existing function/procedure


- Select the function/procedure for editing in the navigation pane/object pane.
- Right-click and select the  **Design Function** from the popup menu or simply double-click the function/ procedure.  
or
- Click the  **Design Function** from the object pane toolbar.
- Edit function/procedure properties on the appropriate tabs of the Function/Procedure Designer.

## Run Function/Procedure

To run a function/procedure in the navigation pane/object pane



- Select the function/procedure for executing in the navigation pane/object pane.
- Click the  **Execute Function** from the object pane toolbar.  
or
- Right-click and select  **Execute Function** from the popup menu.
- View the returned data on the Result tab.

To run a function/procedure in the Function/Procedure Designer

- Create a new function/procedure or open the existing function/procedure.
- Click  **Run**.
- View the returned data on the Result tab.

## Delete Function/Procedure

To delete a function/procedure


- Select the function/procedure for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Function** from the popup menu.  
or
- Click the  **Delete Function** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Function/Procedure Information

To achieve a function/procedure information

- Select the function/procedure in the navigation pane/object pane.
- Right-click the selected function/procedure and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQL Server Function Wizard

Click the  **New Function** from the object pane toolbar. The **Function Wizard** will pop up and it allows you to create a procedure/function easily.

- [Setting Routine Type](#)
- [Setting Parameters for Procedure/Function](#)
- [Setting Return Type for Function](#)
- [Setting Options for Procedure/Function](#)

You are allowed not to show the **Function Wizard** when create new procedure/function.

**Hint:** Once uncheck the **Show wizard next time**, you can go to Options to enable it.

## Setting SQL Server Routine Type

Specify the **Name** of the routine.

Select the type of the routine: **Procedure** or **Function**

## Setting Parameters for SQL Server Procedure/Function

### Procedure

Define the parameter(s) of the procedure. Set the parameter **Name**, **Type Schema**, **Type**, **Default Value**, **Output** and **Read Only** under corresponding columns.

### Function

Define the parameter(s) of the function. Set the parameter **Name**, **Type Schema**, **Type**, **Default Value** and **Read Only** under corresponding columns.

## Setting Return Type for SQL Server Function

Select the **Function type** from the dropdown list.

Then, choose the **Schema** and the **Return Type** from the list if necessary.

**Note:** Only function supports return type.

## Setting Options for SQL Server Procedure/Function

### ☒ **Encryption**

The database will convert the CREATE statement to an obfuscated format.

**Note:** SQL Azure does not support.

### ☒ **Recompile**

Instruct the database not to cache a plan for the procedure.

**Note:** Available only for procedure.

### ☒ **Schema binding**

The function is bound to the database objects that it references.

**Note:** Available only for function.

### ☒ **NULL on NULL Input**

Indicate the server can return null without invoking the function body.

**Note:** Available only for function and support from SQL Server 2005 or later.

### ☒ **Execute As**

Specify the runtime user of the function.

**Note:** Support from SQL Server 2005 or later and SQL Azure.

### ☒ **For replication**

Stored procedures created for replication cannot be executed on the Subscriber.

**Note:** Available only for procedure and SQL Azure does not support.



## SQL Server Function/Procedure Designer

**Function/Procedure Designer** allows you to edit the existing function/procedure definition and more.







- [Editing Function/Procedure Definition](#)
- Editing Function/Procedure Comment (SQL Azure does not support)
- Function/Procedure SQL Preview
- [Viewing Function/Procedure Result](#)

## Editing SQL Server Function/Procedure Definition

Edit the function/procedure definition under the **Definition** tab.

The **Code Outline** window displays information about the function/procedure including parameter, code body, etc. To show the **Code Outline** window, simply choose View -> **Code Outline**.

**Note:** Available only in Full Version.


	Refresh the code outline.
	Show the detail view of the code outline.
	Turn mouse over highlight on or off.
	Expand the selected item.
	Collapse the selected item.
	Toggle sorting by position.

Example:

```
CREATE PROCEDURE [schemaname].
AS
BEGIN
    -- routine body goes here, e.g.
    -- SELECT 'Navicat for SQL Server'
END
```

**Hint:** To customize the view of the editor and find out more features for sql editing, see Editor View and More Features.


## Viewing SQL Server Function/Procedure Result

To run the function/procedure click  **Run** on the toolbar. If the SQL statement is correct, the statement will be executed and, if the statement is supposed to return data, the **Message** and **Result** tabs open with the message log and data returned by the function/procedure. If an error occurs while executing the function/procedure, execution stops, the appropriate error message is displayed.

If the function/procedure requires input parameter, the **Input Parameters** box will popup.



## SQL Server Indexes

An index in a database lets you quickly find specific information in a table or indexed view. An index contains keys built from one or more columns in the table, or view, and pointers that map to the storage location of the specified data. You can significantly improve the performance of database queries and applications by creating well-designed indexes to support your queries. Indexes can reduce the amount of data that must be read to return the query result set. Indexes can also enforce uniqueness on the rows in a table, ensuring the data integrity of the table data.




Just simply click  -> **Index** to open an object pane for **Index**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected Index.

### Create Index

To create a new index



- Select anywhere on the object pane.
- Click the  **New Index** from the object pane toolbar.  
or
- Right-click and select  **New Index** from the popup menu.
- Edit index properties on the appropriate tabs of the Index Designer.

To create a new index with modification as one of the existing index

- Select the index for modifying in the object pane.
- Right-click and select the  **Design Index** from the popup menu or simply double-click the index.  
or
- Click the  **Design Index** from the object pane toolbar.
- Modify index properties on the appropriate tabs of the Index Designer.
- Click  **Save As**.

## Edit Index

To edit the existing index (manage its properties etc)

- Select the index for editing in the object pane.
- Right-click and select the  **Design Index** from the popup menu or simply double-click the index.  
or
- Click the  **Design Index** from the object pane toolbar.
- Edit index properties on the appropriate tabs of the Index Designer.

To change the name of the index

- Select the index for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.



## Maintain Index

To maintain an index

- Select the index for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Rebuild
  - Reorganize
  - Disable

## Delete Index

To delete an index

- Select the index for deleting in the object pane.
- Right-click and select the  **Delete Index** from the popup menu.  
or
- Click the  **Delete Index** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Index Information

To achieve an index information

- Select the index in the object pane.
- Right-click the selected index and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQL Server Index Designer

**Index Designer** is the basic Navicat tool for working with indexes. It allows you to create new index and edit the existing index properties.

- [Editing Nonclustered Index Properties](#)
- [Editing Clustered Index Properties](#)
- [Editing XML Index Properties](#) (Support from SQL Server 2005 or later)
- [Editing Spatial Index Properties](#) (Support from SQL Server 2008 or later and SQL Azure)
- Editing Index Comment (SQL Azure does not support)
- Index SQL Preview

## Editing SQL Server Nonclustered Index Properties

To create a nonclustered index:

- [Editing Nonclustered Index General](#)
- [Editing Nonclustered Index Filter](#) (Support from SQL Server 2008 or later and SQL Azure)
- [Editing Advanced Nonclustered Index Properties](#)
- [Editing Nonclustered Index Storage](#)



## Editing SQL Server Nonclustered Index General

### Type

Choose the index type: **Nonclustered**, Clustered, XML or Spatial

### ☒ Unique

A unique index is one in which no two rows are permitted to have the same index key value.

### Table / View

Choose to create a table index or a view index.

### Table name or View Name

Select a table or a view.

### Columns

Select the column or columns on which the index is based and the sorting order.

### Included Columns

Select the non-key columns to be added to the leaf level of the nonclustered index.

**Note:** Support from SQL Server 2005 or later and SQL Azure.

## Editing SQL Server Nonclustered Index Filter

To create a filtered index, specify which rows to include in the index.

**Note:** Support from SQL Server 2008 or later and SQL Azure.

### Example:

```
StartDate > '20040101' AND EndDate <= '20040630'
```

## Editing Advanced SQL Server Nonclustered Index Properties

### ☒ Ignore duplicate key values

A warning message will occur when duplicate key values are inserted into a unique index. Only the rows violating the uniqueness constraint will fail.

### ☒ Recompute statistics

Enable automatic statistics updating.

### ☒ Allow row locks

Row locks are allowed when accessing the index. The Database Engine determines when row locks are used.

**Note:** Support from SQL Server 2005 or later.

### ☒ Allow page locks

Page locks are allowed when accessing the index. The Database Engine determines when page locks are used.

**Note:** Support from SQL Server 2005 or later.

## Create / Rebuild Option

### Fill Factor (%)

Specify a percentage that indicates how full the Database Engine should make the leaf level of each index page during index creation or rebuild. Fill Factor must be an integer value from 1 to 100.

**Note:** SQL Azure does not support.

### ☒ Pad Index

The percentage of free space that is specified by fillfactor is applied to the intermediate-level pages of the index.

**Note:** Support from SQL Server 2005 or later.

### ☒ Sort in tempdb

Specify to store temporary sort results in tempdb.

**Note:** SQL Azure does not support.

## **Online**

Long-term table locks are not held for the duration of the index operation.

**Note:** Support from SQL Server 2005 or later and SQL Azure.

## **Max. degree of parallelism**

Override the max degree of parallelism configuration option for the duration of the index operation.

**Note:** Support from SQL Server 2005 or later.

## Editing SQL Server Nonclustered Index Storage

SQL Azure does not support this tab.

### On Filegroup

#### Filegroup

Choose a filegroup.

### On Partition Scheme

**Note:** Support from SQL Server 2005 or later.

#### Partition Scheme

Choose a partition scheme.

#### Partition Column

Choose a partition column name.

### Data Compression

**Note:** Support from SQL Server 2008 or later.

#### Partition Number

The partition which the DATA\_COMPRESSION setting applies.

#### Type

##### **NONE**

Index or specified partitions are not compressed.

##### **ROW**

Index or specified partitions are compressed by using row compression.

##### **PAGE**

Index or specified partitions are compressed by using page compression.

## Editing SQL Server Clustered Index Properties

To create a clustered index:

- [Editing Clustered Index General](#)
- [Editing Advanced Clustered Index Properties](#)
- [Editing Clustered Index Storage](#)

## Editing SQL Server Clustered Index General

### Type

Choose the index type: Nonclustered, **Clustered**, XML or Spatial

### ☒ Unique

A unique index is one in which no two rows are permitted to have the same index key value.

### Table / View

Choose to create a table index or a view index.

### Table name or View Name

Select a table or a view.

### Columns

Select the column or columns on which the index is based and the sorting order.

## Editing Advanced SQL Server Clustered Index Properties

### ☒ **Ignore duplicate key values**

A warning message will occur when duplicate key values are inserted into a unique index. Only the rows violating the uniqueness constraint will fail.

### ☒ **Recompute statistics**

Enable automatic statistics updating.

### ☒ **Allow row locks**

Row locks are allowed when accessing the index. The Database Engine determines when row locks are used.

**Note:** Support from SQL Server 2005 or later.

### ☒ **Allow page locks**

Page locks are allowed when accessing the index. The Database Engine determines when page locks are used.

**Note:** Support from SQL Server 2005 or later.

## Create / Rebuild Option

### **Fill Factor (%)**

Specify a percentage that indicates how full the Database Engine should make the leaf level of each index page during index creation or rebuild. Fill Factor must be an integer value from 1 to 100.

**Note:** SQL Azure does not support.

### ☒ **Pad Index**

The percentage of free space that is specified by fillfactor is applied to the intermediate-level pages of the index.

**Note:** Support from SQL Server 2005 or later.

### ☒ **Sort in tempdb**

Specify to store temporary sort results in tempdb.

**Note:** SQL Azure does not support.



## **Online**

Long-term table locks are not held for the duration of the index operation.

**Note:** Support from SQL Server 2005 or later and SQL Azure.

## **Max. degree of parallelism**

Override the max degree of parallelism configuration option for the duration of the index operation.

**Note:** Support from SQL Server 2005 or later.

## Editing SQL Server Clustered Index Storage

SQL Azure does not support this tab.

### On Filegroup

#### Filegroup

Choose a filegroup.

#### File Stream Filegroup

Choose a filegroup for FILESTREAM data.

**Note:** Support from SQL Server 2008 or later.

### On Partition Scheme

**Note:** Support from SQL Server 2005 or later.

#### Partition Scheme

Choose a partition scheme.

#### Partition Column

Choose a partition column name.

#### File Stream Partition Scheme

Choose a partition scheme for FILESTREAM data.

**Note:** Support from SQL Server 2008 or later.

### Data Compression

**Note:** Support from SQL Server 2008 or later.

#### Partition Number

The partition which the DATA\_COMPRESSION setting applies.

#### Type

##### NONE

Index or specified partitions are not compressed.

**ROW**

Index or specified partitions are compressed by using row compression.

**PAGE**

Index or specified partitions are compressed by using page compression.

## Editing SQL Server XML Index Properties

To create a XML index:

**Note:** Support from SQL Server 2005 or later.

- [Editing XML Index General](#)
- [Editing Advanced XML Index Properties](#)

## Editing SQL Server XML Index General

### Type

Choose the index type: Nonclustered, Clustered, **XML** or Spatial

### Table / View

Must be TABLE.

### Table name

Select a table.

### XML Column

Select the xml column on which the index is based.

### XML Index Type

#### **PRIMARY**

A clustered index is created with the clustered key formed from the clustering key of the user table and an XML node identifier.

#### **PATH secondary**

Create a secondary XML index on columns built on path values and node values in the primary XML index. In the PATH secondary index, the path and node values are key columns that allow efficient seeks when searching for paths.

#### **VALUE secondary**

Create a secondary XML index on columns where key columns are (node value and path) of the primary XML index.

#### **PROPERTY secondary**

Create a secondary XML index on columns (PK, path and node value) of the primary XML index where PK is the primary key of the base table.

### Primary XML Index

Specify the primary XML index to use in creating a secondary XML index.

## Editing Advanced SQL Server XML Index Properties

### ☒ **Recompute statistics**

Enable automatic statistics updating.

### ☒ **Allow row locks**

Row locks are allowed when accessing the index. The Database Engine determines when row locks are used.

### ☒ **Allow page locks**

Page locks are allowed when accessing the index. The Database Engine determines when page locks are used.

## Create / Rebuild Option

### **Fill Factor (%)**

Specify a percentage that indicates how full the Database Engine should make the leaf level of each index page during index creation or rebuild. Fill Factor must be an integer value from 1 to 100.

### ☒ **Pad Index**

The percentage of free space that is specified by fillfactor is applied to the intermediate-level pages of the index.

### ☒ **Sort in tempdb**

Specify to store temporary sort results in tempdb.

### **Max. degree of parallelism**

Override the max degree of parallelism configuration option for the duration of the index operation.

## Editing SQL Server Spatial Index Properties

To create a spatial index:

**Note:** Support from SQL Server 2008 or later and SQL Azure.

- [Editing Spatial Index General](#)
- [Editing Advanced Spatial Index Properties](#)

## Editing SQL Server Spatial Index General

### Type

Choose the index type: Nonclustered, Clustered, XML or **Spatial**

### Table / View

Must be TABLE.

### Table name

Select a table.

### Tessellation Scheme

The tessellation scheme for the spatial index.

### Bounding Box

Specify a numeric four-tuple that defines the four coordinates of the bounding box: the x-min and y-min coordinates of the lower, left corner, and the x-max and y-max coordinates of the upper right corner.

#### Min. Coordinates

Specify the x-coordinate (X) and y-coordinate (Y) of the lower-left corner of the bounding box.

#### Max. Coordinates

Specify the x-coordinate (X) and y-coordinate (Y) of the upper-right corner of the bounding box.

### Grid Density

Define the density of the grid at each level of a tessellation scheme.

#### Level 1

Specify the first (top) level grid.

#### Level 2

Specify the second-level grid.

#### Level 3

Specify the third-level grid.



## **Level 4**

Specify the fourth-level grid.

## **Cells Per Object**

Specify the number of tessellation cells per object that can be used for a single spatial object in the index by the tessellation process.

## Editing Advanced SQL Server Spatial Index Properties

### ☒ **Recompute statistics**

Enable automatic statistics updating.

### ☒ **Allow row locks**

Row locks are allowed when accessing the index. The Database Engine determines when row locks are used.

**Note:** SQL Azure does not support.

### ☒ **Allow page locks**

Page locks are allowed when accessing the index. The Database Engine determines when page locks are used.

**Note:** SQL Azure does not support.

## Create / Rebuild Option

**Note:** SQL Azure does not support.

### **Fill Factor (%)**

Specify a percentage that indicates how full the Database Engine should make the leaf level of each index page during index creation or rebuild. Fill Factor must be an integer value from 1 to 100.

### ☒ **Pad Index**

The percentage of free space that is specified by fillfactor is applied to the intermediate-level pages of the index.

### ☒ **Sort in tempdb**


Specify to store temporary sort results in tempdb.

### **Max. degree of parallelism**

Override the max degree of parallelism configuration option for the duration of the index operation.



## SQL Server Synonyms

A synonym is an alternative name for a schema-scoped object. Client applications can use a single-part name to reference a base object by using a synonym instead of using a two-part, three-part, or four-part name to reference the base object.




Just simply click -> **Synonym** to open an object pane for **Synonym**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected synonym.

### Create Synonym

To create a new synonym



- Select anywhere on the object pane.
- Click the  **New Synonym** from the object pane toolbar.  
or
- Right-click and select  **New Synonym** from the popup menu.
- Edit synonym properties on the appropriate tabs of the Synonym Designer.

To create a new synonym with modification as one of the existing synonym

- Select the synonym for modifying in the object pane.
- Right-click and select the  **Design Synonym** from the popup menu or simply double-click the synonym.  
or
- Click the  **Design Synonym** from the object pane toolbar.
- Modify synonym properties on the appropriate tabs of the Synonym Designer.
- Click  **Save As**.

### Edit Synonym

To edit the existing synonym(manage its general etc)



- Select the synonym for editing in the object pane.
- Right-click and select the  **Design Synonym** from the popup menu or simply double-click the synonym.  
or
- Click the  **Design Synonym** from the object pane toolbar.
- Edit synonym properties on the appropriate tabs of the Synonym Designer.

To change the name of the synonym

- Select the synonym for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

## Delete Synonym

To delete a synonym

- Select the synonym for deleting in the object pane.
- Right-click and select the  **Delete Synonym** from the popup menu.  
or
- Click the  **Delete Synonym** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Synonym Information

To achieve a synonym information

- Select the synonym in the object pane.
- Right-click the selected synonym and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQL Server Synonym Designer

**Synonym Designer** is the basic Navicat tool for working with synonym. It allows you to create new synonym and edit the existing synonym properties.

- [Editing Synonym General](#)
- Editing Synonym Comment (SQL Azure does not support)
- Synonym SQL Preview

## Editing SQL Server Synonym General

### Object Linked Server

The name of the server on which base object is located.

**Note:** SQL Azure does not support.

### Object Database

The name of the database in which the base object is located.

### Object Schema

The name of the schema of the base object.

### Object Type

The object type.


### Object

The name of the base object that the synonym references.

## SQL Server Triggers



A trigger is a special kind of stored procedure that automatically executes when an event occurs in the database server.

See [Triggers](#) for details.




Just simply click  -> **Trigger** to open an object pane for **Trigger**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected trigger.

### Create Trigger

To create a new trigger



- Select anywhere on the object pane.
- Click the  **New Trigger** from the object pane toolbar.  
or
- Right-click and select  **New Trigger** from the popup menu.
- Edit trigger properties on the appropriate tabs of the Trigger Designer.

To create a new trigger with modification as one of the existing trigger

- Select the trigger for modifying in the object pane.
- Right-click and select the  **Design Trigger** from the popup menu or simply double-click the trigger.  
or
- Click the  **Design Trigger** from the object pane toolbar.
- Modify trigger properties on the appropriate tabs of the Trigger Designer.
- Click  **Save As**.

## Edit Trigger

To edit the existing trigger (manage its general, advance, etc)

- Select the trigger for editing in the object pane.
- Right-click and select the  **Design Trigger** from the popup menu or simply double-click the trigger.  
or
- Click the  **Design Trigger** from the object pane toolbar.
- Edit trigger properties on the appropriate tabs of the Trigger Designer.

To change the name of the trigger

- Select the trigger for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.



## Maintain Trigger

To maintain a trigger

- Select the trigger for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Enable
  - Disable

## Delete Trigger

To delete a trigger

- Select the trigger for deleting in the object pane.
- Right-click and select the  **Delete Trigger** from the popup menu.  
or
- Click the  **Delete Trigger** from the object pane toolbar.
- Confirm deleting in the dialog window.



## Achieve Trigger Information

To achieve a trigger information

- Select the trigger in the object pane.
- Right-click the selected trigger and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQL Server Trigger Designer

**Trigger Designer** is the basic Navicat tool for working with triggers. It allows you to create new trigger and edit the existing trigger definition.

- [Editing Trigger General](#)
- [Setting Advanced Trigger Properties](#)
- [Editing Trigger Definition](#)
- Editing Trigger Comment (SQL Azure does not support)
- Trigger SQL Preview

## Editing SQL Server Trigger General

### Trigger Type

Choose Table or View on which the DML trigger is executed.

#### ☒ **Enable**

Check this option to enable the trigger.

### Table name or View name

Choose a table or a view.

### Fire

#### **AFTER**

Specify that the DML trigger is fired only when all operations specified in the triggering SQL statement have executed successfully.

#### **INSTEAD OF**

Specify that the DML trigger is executed instead of the triggering SQL statement, therefore, overriding the actions of the triggering statements.

### On Event

#### ☒ **Insert**

The trigger is activated whenever a new row is inserted into the table.

#### ☒ **Delete**

The trigger is activated whenever a row is deleted from the table.

#### ☒ **Update**

The trigger is activated whenever a row is modified.

### Definition Type

Choose the definition type.

**Note:** Support from SQL Server 2005 or later.

## Setting Advanced SQL Server Trigger Properties

### **Execute As**

Specify the security context under which the trigger is executed.

**Note:** Support from SQL Server 2005 or later and SQL Azure.

### ☒ **Encrypted**

Obfuscate the text of the CREATE TRIGGER statement.

**Note:** Support from SQL Server 2005 or later.

### ☒ **Not For Replication**

Indicate that the trigger should not be executed when a replication agent modifies the table that is involved in the trigger.

**Note:** SQL Azure does not support.

### ☒ **With Append**

Specify that an additional trigger of an existing type should be added.

**Note:** SQL Azure does not support.

## Editing SQL Server Trigger Definition

The **Definition** tab allows you to edit valid SQL or procedure statements in the trigger definition.


**Note:** This tab will appear when the **Definition Type** is set to **SQL Statement** in General tab or when connecting to SQL Azure.

## SQL Server Linked Servers

A linked server configuration enables SQL Server to execute commands against OLE DB data sources on remote servers. Linked servers offer the following advantages:



- Remote server access.
- The ability to issue distributed queries, updates, commands, and transactions on heterogeneous data sources across the enterprise.
- The ability to address diverse data sources similarly.

**Note:** SQL Azure does not support.




Just simply click  -> **Linked Server** to open an object pane for **Linked Server**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected linked server.

### Create Linked Server

To create a new linked server



- Select anywhere on the object pane.
- Click the  **New Linked Server** from the object pane toolbar.  
or
- Right-click and select  **New Linked Server** from the popup menu.
- Edit linked server properties on the appropriate tabs of the Linked Server Designer.

To create a new linked server with modification as one of the existing linked server

- Select the linked server for modifying in the object pane.
- Right-click and select the  **Design Linked Server** from the popup menu or simply double-click the linked server.  
or
- Click the  **Design Linked Server** from the object pane toolbar.
- Modify linked server properties on the appropriate tabs of the Linked Server Designer.
- Click  **Save As**.



## Edit Linked Server

To edit the existing linked server(manage its general etc)

- Select the linked server for editing in the object pane.
- Right-click and select the  **Design Linked Server** from the popup menu or simply double-click the linked server.  
or
- Click the  **Design Linked Server** from the object pane toolbar.
- Edit linked server properties on the appropriate tabs of the Linked Server Designer.

## Delete Linked Server

To delete a linked server

- Select the linked server for deleting in the object pane.
- Right-click and select the  **Delete Linked Server** from the popup menu.  
or
- Click the  **Delete Linked Server** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Linked Server Information

To achieve a linked server information

- Select the linked server in the object pane.
- Right-click the selected linked server and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQL Server Linked Server Designer

**Linked Server Designer** is the basic Navicat tool for working with linked servers. It allows you to create new linked server and edit the existing linked server properties.

- [Editing Linked Server General](#)
- [Editing Linked Server Security](#)
- [Setting Advanced Linked Server Properties](#)
- Linked Server SQL Preview



## Editing SQL Server Linked Server General

### Server Type

Choose the server type: **SQL Server** or **Other Data Source**

If you choose **Other Data Source**, define the required information.

#### Provider

Choose the unique programmatic identifier (PROGID) of the OLE DB provider corresponding to the data source.

#### Product Name

Define the product name of the OLE DB data source to add as a linked server.

#### Data Source

Define the name of the data source as interpreted by the OLE DB provider.

#### Provider String

Define the OLE DB provider-specific connection string that identifies a unique data source.

#### Location

Define the location of the database as interpreted by the OLE DB provider.

#### Catalog

Define the catalog to be used when making a connection to the OLE DB provider.

## Editing SQL Server Linked Server Security

In this tab, add or delete a mapping between logins on the local instance of SQL Server and remote logins on the linked server.

### Local Login

Choose a login on the local server.

#### ☒ Impersonate

Check this option to specify that logins use their own credentials to connect to the linked server.

### Remote Login

Enter the username used to connect the linked server.

### Remote Password

Enter the user password.

Set the action when a login not defined in the list:

- Not be made
  - Be made without using a security context
  - Be made using the login's current security context
  - Be made using the following security context
- Set the Remote login and Password

## Setting Advanced SQL Server Linked Server Properties

### Connect Timeout

Define the time-out value for connecting to a linked server. If 0, use the sp\_configure default.

### Query Timeout

Define the time-out value for queries against a linked server. If 0, use the sp\_configure default.

#### ☒ Data Access

Check this option to enable a linked server for distributed query access.

#### ☒ Collation Compatible

If this option is checked, SQL Server assumes that all characters in the linked server are compatible with the local server, with regard to character set and collation sequence (or sort order). This enables SQL Server to send comparisons on character columns to the provider.

#### ☒ Use Remote Collation

If this option is checked, the collation of remote columns is used for SQL Server data sources, and the collation specified in collation name is used for non-SQL Server data sources.

### Collation

Specify the name of the collation used by the remote data source if Use Remote Collation is checked and the data source is not a SQL Server data source. The name must be one of the collations supported by SQL Server.

#### ☒ Lazy Schema Validation

If this option is checked, skip schema checking of remote tables at the beginning of the query.

#### ☒ Publisher

Check this option to enable publisher.

#### ☒ Subscriber

Check this option to enable subscriber.

#### ☒ Distributor

Check this option to enable distributor.

## ☒ **RPC**

Check this option to enable RPC from the given server.

## ☒ **RPC Out**

Check this option to enable RPC to the given server.

## ☒ **Promotion of Distributed Transactions for RPC**


Use this option to protect the actions of a server-to-server procedure through a Microsoft Distributed Transaction Coordinator (MS DTC) transaction.

**Note:** Support from SQL Server 2005 or later.

## SQL Server Server Triggers



A server trigger can be a DDL or logon trigger for current server. DDL triggers execute in response to a variety of data definition language (DDL) events. These events primarily correspond to Transact-SQL CREATE, ALTER, and DROP statements, and certain system stored procedures that perform DDL-like operations. Logon triggers fire in response to the LOGON event that is raised when a user sessions is being established.

**Note:** Support from SQL Server 2005 or later.




Just simply click -> **Server Trigger** to open an object pane for **Server Trigger**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected server trigger.

### Create Server Trigger

To create a new server trigger



- Select anywhere on the object pane.
- Click the  **New Server Trigger** from the object pane toolbar.  
or
- Right-click and select  **New Server Trigger** from the popup menu.
- Edit server trigger properties on the appropriate tabs of the Server Trigger Designer.

To create a new server trigger with modification as one of the existing server trigger

- Select the server trigger for modifying in the object pane.
- Right-click and select the  **Design Server Trigger** from the popup menu or simply double-click the server trigger.  
or
- Click the  **Design Server Trigger** from the object pane toolbar.
- Modify server trigger properties on the appropriate tabs of the Server Trigger Designer.
- Click  **Save As**.

## Edit Server Trigger

To edit the existing server trigger (manage its general etc)

- Select the server trigger for editing in the object pane.
- Right-click and select the  **Design Server Trigger** from the popup menu or simply double-click the server trigger.  
or
- Click the  **Design Server Trigger** from the object pane toolbar.
- Edit server trigger properties on the appropriate tabs of the Server Trigger Designer.



## Maintain Server Trigger

To maintain a server trigger

- Select the server trigger for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Enable
  - Disable

## Delete Server Trigger

To delete a server trigger

- Select the server trigger for deleting in the object pane.
- Right-click and select the  **Delete Server Trigger** from the popup menu.  
or
- Click the  **Delete Server Trigger** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Server Trigger Information

To achieve a server trigger information

- Select the server trigger in the object pane.
- Right-click the selected server trigger and choose **Server Trigger Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQL Server Server Trigger Designer

**Server Trigger Designer** is the basic Navicat tool for working with server triggers. It allows you to create new server trigger and edit the existing server trigger definition.

- [Editing Server Trigger General](#)
- [Setting Advanced Server Trigger Properties](#)
- [Editing Server Trigger Definition](#)
- Server Trigger SQL Preview

## Editing SQL Server Server Trigger General

### Trigger Type

Choose the trigger type.

#### ☒ **Enable**

Check this option to enable the trigger.

### Definition Type

Choose the definition type.

### Events

Check the DDL event form the list.



## Setting Advanced SQL Server Server Trigger Properties

### Execute As

Specify the security context under which the trigger is executed.

### ☒ Encrypted

Obfuscate the text of the CREATE TRIGGER statement.

## Editing SQL Server Server Trigger Definition


The **Definition** tab allows you to edit valid SQL or procedure statements in the server trigger definition.

**Note:** This tab will appear when the **Definition Type** is set to **SQL statements** in General tab.

## SQL Server Assemblies



An assembly is a managed application module that contains class metadata and managed code as an object in an instance of SQL Server. By referencing this module, common language runtime (CLR) functions, stored procedures, triggers, user-defined aggregates, and user-defined types can be created in the database.

**Note:** Support from SQL Server 2005 or later.




Just simply click  -> **Assembly** to open an object pane for **Assembly**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected assembly.

### Create Assembly

To create a new assembly



- Select anywhere on the object pane.
- Click the  **New Assembly** from the object pane toolbar.  
or
- Right-click and select  **New Assembly** from the popup menu.
- Edit assembly properties on the appropriate tabs of the Assembly Designer.

To create a new assembly with modification as one of the existing assembly

- Select the assembly for modifying in the object pane.
- Right-click and select the  **Design Assembly** from the popup menu or simply double-click the assembly.  
or
- Click the  **Design Assembly** from the object pane toolbar.
- Modify assembly properties on the appropriate tabs of the Assembly Designer.
- Click  **Save As**.

## Edit Assembly

To edit the existing assembly(manage its general etc)

- Select the assembly for editing in the object pane.
- Right-click and select the  **Design Assembly** from the popup menu or simply double-click the assembly.  
or
- Click the  **Design Assembly** from the object pane toolbar.
- Edit assembly properties on the appropriate tabs of the Assembly Designer.



## Maintain Assembly

To maintain an assembly

- Select the assembly for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Set Visible
  - Set Invisible

## Delete Assembly

To delete an assembly

- Select the assembly for deleting in the object pane.
- Right-click and select the  **Delete Assembly** from the popup menu.  
or
- Click the  **Delete Assembly** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Assembly Information

To achieve an assembly information

- Select the assembly in the object pane.
- Right-click the selected assembly and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQL Server Assembly Designer

**Assembly Designer** is the basic Navicat tool for working with assemblies. It allows you to create new assembly and edit the existing assembly properties.

- [Editing Assembly General](#)
- Editing Assembly Comment
- Assembly SQL Preview

## Editing SQL Server Assembly General

### Owner

Specify the name of a user or role as owner of the assembly.

### Permission set

Specify a set of code access permissions that are granted to the assembly when it is accessed by SQL Server. If not specified, SAFE is applied as the default.

### Assembly

Specify the local path or network location where the assembly that is being uploaded is located, and also the manifest file name that corresponds to the assembly.


### Dependent Assemblies

Uploads a file to be associated with the assembly, such as source code, debug files or other related information, into the server and made visible in the sys.assembly\_files catalog view.

## SQL Server Database Triggers



A database trigger is a DDL trigger to the current database. DDL triggers execute in response to a variety of data definition language (DDL) events. These events primarily correspond to Transact-SQL CREATE, ALTER, and DROP statements, and certain system stored procedures that perform DDL-like operations.

**Note:** Support from SQL Server 2005 or later and SQL Azure.




Just simply click  -> **Database Trigger** to open an object pane for **Database Trigger**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected database trigger.

### Create Database Trigger

To create a new database trigger



- Select anywhere on the object pane.
- Click the  **New Database Trigger** from the object pane toolbar.  
or
- Right-click and select  **New Database Trigger** from the popup menu.
- Edit database trigger properties on the appropriate tabs of the Database Trigger Designer.

To create a new database trigger with modification as one of the existing database trigger

- Select the database trigger for modifying in the object pane.
- Right-click and select the  **Design Database Trigger** from the popup menu or simply double-click the database trigger.  
or
- Click the  **Design Database Trigger** from the object pane toolbar.
- Modify database trigger properties on the appropriate tabs of the Database Trigger Designer.
- Click  **Save As**.

## Edit Database Trigger

To edit the existing database trigger(manage its general etc)

- Select the database trigger for editing in the object pane.
- Right-click and select the  **Design Database Trigger** from the popup menu or simply double-click the database trigger.  
or
- Click the  **Design Database Trigger** from the object pane toolbar.
- Edit database trigger properties on the appropriate tabs of the Database Trigger Designer.



## Maintain Database Trigger

To maintain a database trigger

- Select the database trigger for maintaining in the object pane.
- Right-click and select the **Maintain** from the popup menu.
  - Enable
  - Disable

## Delete Database Trigger

To delete an database trigger

- Select the database trigger for deleting in the object pane.
- Right-click and select the  **Delete Database Trigger** from the popup menu.  
or
- Click the  **Delete Database Trigger** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Database Trigger Information

To achieve a database trigger information

- Select the database trigger in the object pane.
- Right-click the selected database trigger and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.



## SQL Server Database Trigger Designer

**Database Trigger Designer** is the basic Navicat tool for working with database triggers. It allows you to create new database trigger and edit the existing database trigger definition.

- [Editing Database Trigger General](#)
- [Setting Advanced Database Trigger Properties](#)
- [Editing Database Trigger Definition](#)
- Editing Database Trigger Comment (SQL Azure does not support)
- Database Trigger SQL Preview

## Editing SQL Server Database Trigger General

### Trigger Type

Trigger type must be Database Trigger.

### ☒ Enable

Check this option to enable the trigger.

### Definition Type

Choose the definition type.

**Note:** SQL Azure does not support.

### Events

Check the DDL event form the list.

## Setting Advanced SQL Server Database Trigger Properties

### Execute As

Specify the security context under which the trigger is executed.

### ☒ Encrypted

Obfuscate the text of the CREATE TRIGGER statement.

**Note:** SQL Azure does not support.

## Editing SQL Server Database Trigger Definition


The **Definition** tab allows you to edit valid SQL or procedure statements in the database trigger definition.

**Note:** This tab will appear when the **Definition Type** is set to **SQL Statements** in General tab or when connecting to SQL Azure.

## SQL Server Partition Functions



A partition function is a function in the current database that maps the rows of a table or index into partitions based on the values of a specified column.

**Note:** Support from SQL Server 2005 or later.




Just simply click  -> **Partition Function** to open an object pane for **Partition Function**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected partition function.

### Create Partition Function

To create a new partition function



- Select anywhere on the object pane.
- Click the  **New Partition Function** from the object pane toolbar.  
or
- Right-click and select  **New Partition Function** from the popup menu.
- Edit partition function properties on the appropriate tabs of the Partition Function Designer.

To create a new partition function with modification as one of the existing partition function

- Select the partition function for modifying in the object pane.
- Right-click and select the  **Design Partition Function** from the popup menu or simply double-click the partition function.  
or
- Click the  **Design Partition Function** from the object pane toolbar.
- Modify partition function properties on the appropriate tabs of the Partition Function Designer.
- Click  **Save As**.



## Edit Partition Function

To edit the existing partition function(manage its general etc)

- Select the partition function for editing in the object pane.
- Right-click and select the  **Design Partition Function** from the popup menu or simply double-click the partition function.  
or
- Click the  **Design Partition Function** from the object pane toolbar.
- Edit partition function properties on the appropriate tabs of the Partition Function Designer.

## Delete Partition Function

To delete an partition function

- Select the partition function for deleting in the object pane.
- Right-click and select the  **Delete Partition Function** from the popup menu.  
or
- Click the  **Delete Partition Function** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Partition Function Information

To achieve an partition function information

- Select the partition function in the object pane.
- Right-click the selected partition function and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQL Server Partition Function Designer

**Partition Function Designer** is the basic Navicat tool for working with partition functions. It allows you to create new partition function.

- [Editing Partition Function General](#)
- Editing Partition Function Comment
- Partition Function SQL Preview

## Editing SQL Server Partition Function General

### Input Parameter Type

Choose the data type of the column used for partitioning.

### Length

Specify the length of the data type if necessary.

### Decimals

Specify the decimals of the data type if necessary.

### Collation

Specify the collation of the data type if necessary.

### ☒ Boundary values belong to right interval

Specify to the right of each boundary value interval.

### Boundary Values


Specify the boundary values for each partition of a partitioned table or index that uses `partition_function_name`.



## SQL Server Partition Schemes



A partition scheme is a scheme in the current database that maps the partitions of a partitioned table or index to filegroups. The number and domain of the partitions of a partitioned table or index are determined in a partition scheme.

**Note:** Support from SQL Server 2005 or later.




Just simply click  -> **Partition Scheme** to open an object pane for **Partition Scheme**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected partition scheme.

### Create Partition Scheme

To create a new partition scheme



- Select anywhere on the object pane.
- Click the  **New Partition Scheme** from the object pane toolbar.  
or
- Right-click and select  **New Partition Scheme** from the popup menu.
- Edit partition scheme properties on the appropriate tabs of the Partition Scheme Designer.

To create a new partition scheme with modification as one of the existing partition scheme

- Select the partition scheme for modifying in the object pane.
- Right-click and select the  **Design Partition Scheme** from the popup menu or simply double-click the partition scheme.  
or
- Click the  **Design Partition Scheme** from the object pane toolbar.
- Modify partition scheme properties on the appropriate tabs of the Partition Scheme Designer.
- Click  **Save As**.



## Edit Partition Scheme

To edit the existing partition scheme(manage its general etc)

- Select the partition scheme for editing in the object pane.
- Right-click and select the  **Design Partition Scheme** from the popup menu or simply double-click the partition scheme.  
or
- Click the  **Design Partition Scheme** from the object pane toolbar.
- Edit partition scheme properties on the appropriate tabs of the Partition Scheme Designer.

## Delete Partition Scheme

To delete an partition scheme

- Select the partition scheme for deleting in the object pane.
- Right-click and select the  **Delete Partition Scheme** from the popup menu.  
or
- Click the  **Delete Partition Scheme** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Partition Scheme Information

To achieve an partition scheme information

- Select the partition scheme in the object pane.
- Right-click the selected partition scheme and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## SQL Server Partition Scheme Designer

**Partition Scheme Designer** is the basic Navicat tool for working with partition functions. It allows you to create new partition scheme.

- [Editing Partition Scheme General](#)
- Editing Partition Scheme Comment
- Partition Scheme SQL Preview

## Editing SQL Server Partition Scheme General

### Partition Function



Choose the partition function.

### Filegroup Mapping

Specify the filegroups to hold the partitions specified by partition\_function\_name.

## SQL Previewer

The **SQL Preview** tab shows the CREATE statement and necessary SQL statements of the database or schema object.

For some database or schema objects, you can use the below dropdown list to show the SQL which will be run when pressing  **Save** or  **Save As** button.

## Table Viewer

**Table Viewer** displays the table data as a grid. Data can be displayed in three modes:  **Grid View**,  **Form View** and **Text/Blob/BFile View**.

The toolbars of Table Viewer provides the following functions for managing data:

- **Commit**

Make permanent all changes performed in the transaction.

**Hint:** The Commit button is visible only when **Auto Commit** is disabled under Option Settings.

**Note:** Available only for Oracle, SQLite and SQL Server.

- **Rollback**

Undo work done in the current transaction.

**Hint:** The Rollback button is visible only when **Auto Commit** is disabled under Option Settings.

**Note:** Available only for Oracle, SQLite and SQL Server.

- **Import Data**

Import data from TXT, CSV, XML, DBF, MS Excel, MS Access, ODBC and more.

- **Export Data**

Export data to MS Word, MS Excel, MS Access, TXT, DBF, HTML, SQL, RTF and more.

- [Filter Data](#)

Allow you to filter records by creating and applying filter criteria for the data grid.

- [Edit TEXT/BLOB/BFile](#)

Allow you to view and edit the content of TEXT, BLOB and BFile fields.

**Note:** Only Oracle supports BFile.

EMPLOYEES @HR (Basic Connection)

File Edit View Window Help

Commit Rollback Import Wizard Export Wizard Filter Wizard Grid View

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
100	Steven	King	SKING	515.123.4567	1987-06-17 00:00:00	AD_PRES
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21 00:00:00	AD_VP
102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13 00:00:00	AD_VP
103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03 00:00:00	IT_PROG
104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21 00:00:00	IT_PROG
105	David	Austin	DAUSTIN	590.423.4569	1997-06-25 00:00:00	IT_PROG
106	Valli	Pataballa	VPATABAL	590.423.4560	1998-02-05 00:00:00	IT_PROG
107	Diana	Lorentz	DLORENTZ	590.423.5567	1999-02-07 00:00:00	IT_PROG
108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17 00:00:00	FI_MGR
109	Daniel	Faviet	DFAVIET	515.124.4169	1994-08-16 00:00:00	FI_ACCC
110	John	Chen	JCHEN	515.124.4269	1997-09-28 00:00:00	FI_ACCC
111	Ismael	Sciarra	ISCIARRA	515.124.4369	1997-09-30 00:00:00	FI_ACCC
112	Jose Manuel	Urman	JMURMAN	515.124.4469	1998-03-07 00:00:00	FI_ACCC
113	Luis	Popp	LPOPP	515.124.4567	1999-12-07 00:00:00	FI_ACCC
114	Den	Raphaely	DRAPHEAL	515.127.4561	1994-12-07 00:00:00	PU_MAN
115	Alexander	Khoo	AKHOO	515.127.4562	1995-05-18 00:00:00	PU_CLER
116	Shelli	Baida	SBAIDA	515.127.4563	1997-12-24 00:00:00	PU_CLER
117	Sigal	Tobias	STOBIAS	515.127.4564	1997-07-24 00:00:00	PU_CLER
118	Guy	Himuro	GHIMURO	515.127.4565	1998-11-15 00:00:00	PU_CLER
119	Karen	Colmenares	KCOLMENA	515.127.4566	1999-08-10 00:00:00	PU_CLER
120	Matthew	Weiss	MWEISS	650.123.1234	1996-07-18 00:00:00	ST_MAN
121	Adam	Fripp	AFRIPP	650.123.2234	1997-04-10 00:00:00	ST_MAN

SELECT \* FROM (SELECT "NAVICAT\_TABLE".\*, ROWNUM "NAVI" Record 1 of 107 in Page 1

## Data View

This topic shows you how you can view and edit data from tables in the simplest and the most direct way.

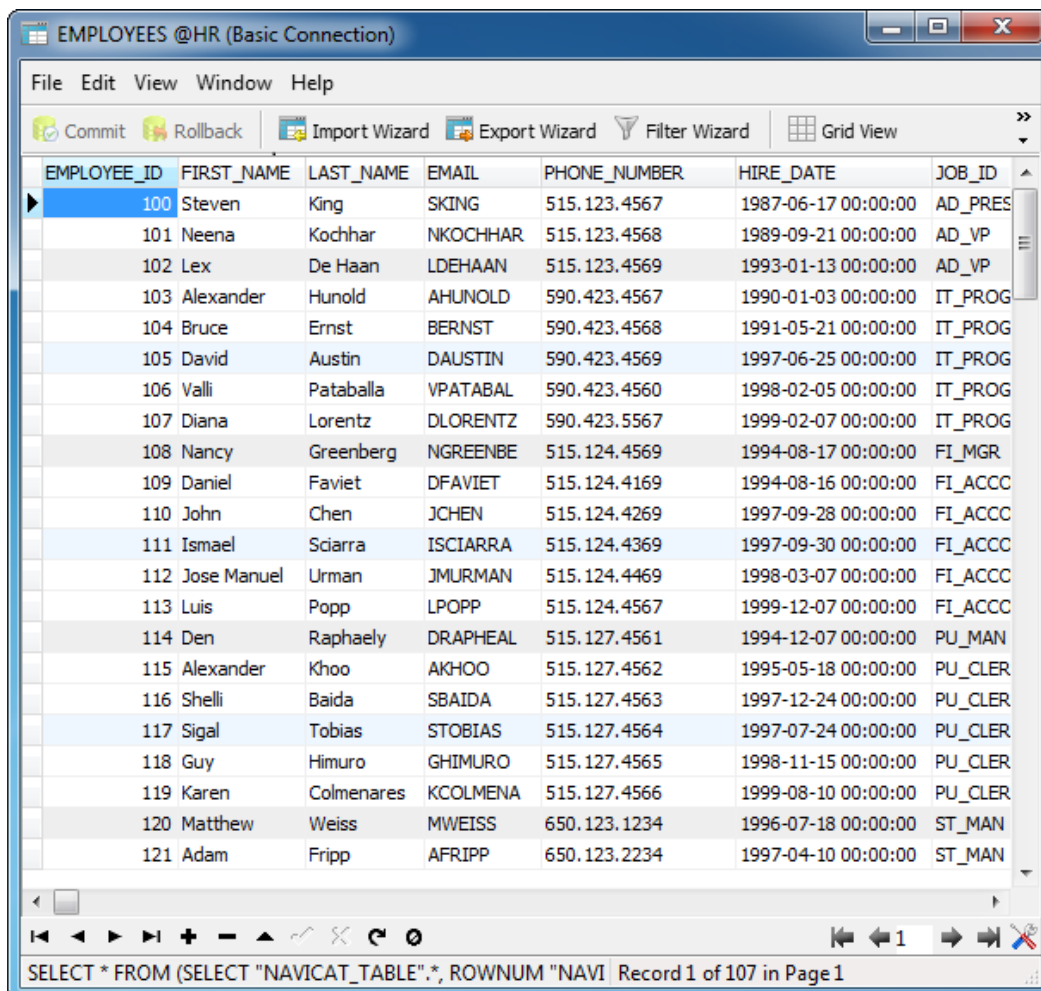
- [Grid View](#)
- [Form View](#)
- [Text/Blob/BFile View](#)



## Grid View

The grid view allows you to view, update, insert, or delete data in a table. The popup menu of the grid provides the following additional functions: set the field value as Null/Empty String, use current field value as a filter, format grid view, and more.

- [Using Navigation Bar](#)
- [Editing Records](#)
- [Sorting and Finding Records](#)
- [Filtering Records](#)
- [Manipulating Raw Data](#) (Available only for MySQL, PostgreSQL, SQLite and SQL Server)
- [Formatting Table Grid](#)














The screenshot shows the Navicat Grid View for the 'EMPLOYEES @HR (Basic Connection)' table. The interface includes a menu bar (File, Edit, View, Window, Help), a toolbar with buttons for Commit, Rollback, Import Wizard, Export Wizard, Filter Wizard, and Grid View, and a table of employee data. The table has columns: EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, and JOB\_ID. The data is displayed in a grid with alternating row colors. At the bottom, there is a navigation bar with various icons and a status bar showing the SQL query 'SELECT \* FROM (SELECT "NAVICAT\_TABLE".\*, ROWNUM "NAVI' and 'Record 1 of 107 in Page 1'.

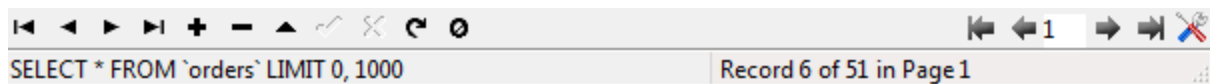
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
100	Steven	King	SKING	515.123.4567	1987-06-17 00:00:00	AD_PRES
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21 00:00:00	AD_VP
102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13 00:00:00	AD_VP
103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03 00:00:00	IT_PROG
104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21 00:00:00	IT_PROG
105	David	Austin	DAUSTIN	590.423.4569	1997-06-25 00:00:00	IT_PROG
106	Valli	Pataballa	VPATABAL	590.423.4560	1998-02-05 00:00:00	IT_PROG
107	Diana	Lorentz	DLORENTZ	590.423.5567	1999-02-07 00:00:00	IT_PROG
108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17 00:00:00	FI_MGR
109	Daniel	Faviet	DFAVIET	515.124.4169	1994-08-16 00:00:00	FI_ACCC
110	John	Chen	JCHEN	515.124.4269	1997-09-28 00:00:00	FI_ACCC
111	Ismael	Sciarra	ISCIARRA	515.124.4369	1997-09-30 00:00:00	FI_ACCC
112	Jose Manuel	Urman	JMURMAN	515.124.4469	1998-03-07 00:00:00	FI_ACCC
113	Luis	Popp	LPOPP	515.124.4567	1999-12-07 00:00:00	FI_ACCC
114	Den	Raphaely	DRAPHEAL	515.127.4561	1994-12-07 00:00:00	PU_MAN
115	Alexander	Khoo	AKHOO	515.127.4562	1995-05-18 00:00:00	PU_CLER
116	Shelli	Baida	SBAIDA	515.127.4563	1997-12-24 00:00:00	PU_CLER
117	Sigal	Tobias	STOBIAS	515.127.4564	1997-07-24 00:00:00	PU_CLER
118	Guy	Himuro	GHIMURO	515.127.4565	1998-11-15 00:00:00	PU_CLER
119	Karen	Colmenares	KCOLMENA	515.127.4566	1999-08-10 00:00:00	PU_CLER
120	Matthew	Weiss	MWEISS	650.123.1234	1996-07-18 00:00:00	ST_MAN
121	Adam	Fripp	AFRIPP	650.123.2234	1997-04-10 00:00:00	ST_MAN






## Using Navigation Bar

**Table Viewer** provides a convenient way to navigate among the records/pages using **Record/Page Navigation Bar** buttons. All buttons are used to navigate left and right to the previous or the next records/pages.

Record Objects	Role
	First Record: allows moving to the first record.
	Previous Record: allows moving one record back (if there is one) from the current record.
	Next Record: allows moving one record ahead.
	Last Record: allows moving to the last record.
	Insert Record: used to enter a new record. At any point when you are working with your table in the grid view, click on this button to get a blank display for a record.
	Erase Record: used to delete an existing record.
	Edit Record: used to enter the edit mode.
	Update Record: used to apply the changes.
	Cancel Changes: used to removes all edits made to the current record.
	Refresh: used to refresh the table.
	Stop: used to stop when loading enormous data from server.

**Note:** The SQL statement shows under the Record Objects indicate any statement has just been executed.



Page Objects	Role
	First Page: allows moving to first page.
	Previous Page: allows moving to previous page.
	Next Page: allows moving to next page.
	Last Page: allows moving to last page.
<b>Record a of b in Page c</b>	Record/Page Indicator: displays the numbers representing the selected record and page. a. the selected record. b. number of records in the current page. c. the current page.
	Limit Record Setting: used to set number of records showing on each page.

## Limit Record Setting

Use the **Limit Record Setting**  button to enter to the edit mode.

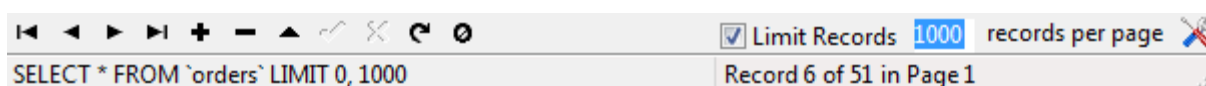
### ☒ **Limit Records**

Check this option if you want to limit the number of records showed on each page. Otherwise, all records will be displayed in one single page.

### ☐ **records per page**

Set the **records per page** value in the edit field. The number representing the number of records showed per page.

**Note:** This setting mode will take effect on current table only. To adjust the global settings, see Options.







## Editing Records

The navigation bar allows you to switch the records quickly, insert, update or delete records. View data as a grid is most helpful for entering new records and editing old records in a table.


### Add Record

To add a record

- Make sure that your cursor is situated in the first blank cell on the table, then enter the desired data. If you are adding the new record into an existing table, just simply click on an existing record and click the  from the navigation bar or press Ctrl+n to get a blank display for a record.
- Watch the graphics symbol in the record selectors box just to the left of your record. It will change from the arrowhead , which indicates that it is the current record, to , which indicates that you are editing this record.
- Just simply move to another record to save the record or click the  from the navigation bar.

### Edit Record


To edit a record

- Select the record that you wish to edit by clicking in the specific field you want to change.
- Type in the new data for that field.
- Just simply move to another record, the new data will overwrite the previous data or click the  from the navigation bar.

**Note:** Close the table is another way to save the records.

### Delete Record

To delete a record

- Select the record that you wish to delete.
- Just simply right-click and select **Delete Record** or click the  from the navigation bar.

## Editing Records with Special Handling

To set **Empty String** for the cell, right-click the selected cell and select **Set to Empty String**.

To set **Null** value for the cell, right-click the selected cell and select **Set to NULL**.

To edit the text field record, just simply click **Show Memo In Grid** from the **View** toolbar.




**Note:** Available only for MySQL, Oracle and PostgreSQL.

CustNo	Company	Addr1
1221	Kauai Dive Shoppe	(WIDEMEMO)
1231	Unisco1	(WIDEMEMO)
1351	Sight Diver00	(WIDEMEMO)
1354	Cayman Divers World Unlimited	(WIDEMEMO)
1356	Tom Sawyer Diving Centre	(WIDEMEMO)


**Hint:** To view/edit the text field record in an ease way, see [Memo Editor](#).

To view images in the grid, just simply click **Show Image In Grid** from the **View** toolbar.

**Note:** Available only for MySQL, Oracle, PostgreSQL and SQL Server.

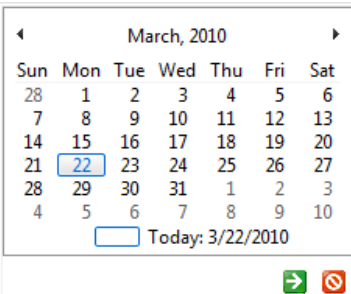
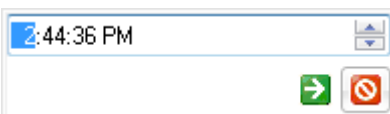
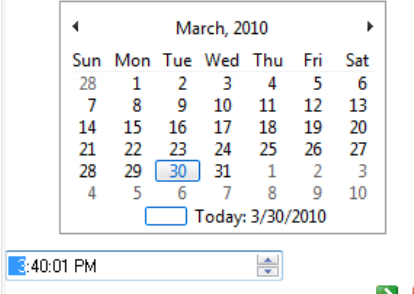
id	image
1	
2	
3	

**Hint:** To view/edit the image in an ease way, see [Image Editor](#).

To edit a **Date/Time** record, just simply click  or press Ctrl+Enter to open the editor for editing.

**Note:** Available only for MySQL, Oracle, PostgreSQL and SQL Server.

- Choose/enter the desired data. The editor used in cell is determined by the field type assigned to the column.

Date	Time	DateTime / Timestamp
		

To edit a **Enum** record, just simply choose the record from the drop-down list.

**Note:** Available only for MySQL.

Student_ID	Subject
1	English
2	English
	Chinese
	IT
	Maths
	Sport

To edit a **Set** record, just simply click  or press Ctrl+Enter to open the editor for editing.

**Note:** Available only for MySQL.

- Select the record(s) from the list. To remove the records, uncheck them in the same way.

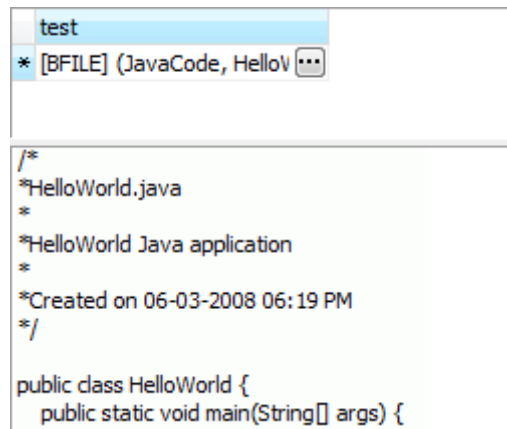
Student ID	Subject
1	English,Chinese
2	English

☐ English  
☐ Chinese  
☒ IT  
☒ Maths  
☐ Sport

To view BFile content, just simply enables **Preview BFile** under the **View** menu.

**Note:** Available only for Oracle.



To generate UUID/GUID, right-click the selected cell and select **Generate UUID**.


**Note:** Available only for PostgreSQL and SQL Server.



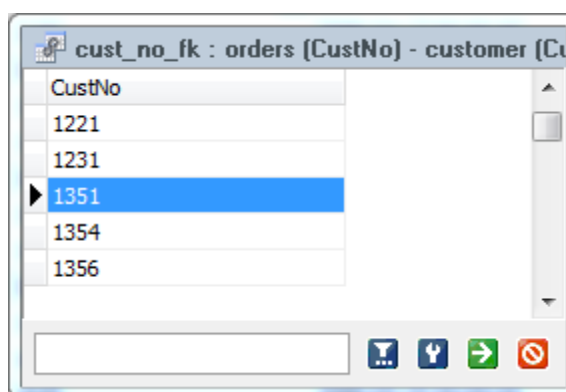
## Editing Records with Foreign Key (Foreign Key Data Selection - Available only in Full Version)

**Foreign Key Data Selection** is a useful tool for letting you to get the available value from the reference table in an easy way. It allows you to show additional record(s) from the reference table and search for a particular record(s).


To include data to the record, just simply click  or press Ctrl+Enter to open the editor for editing.

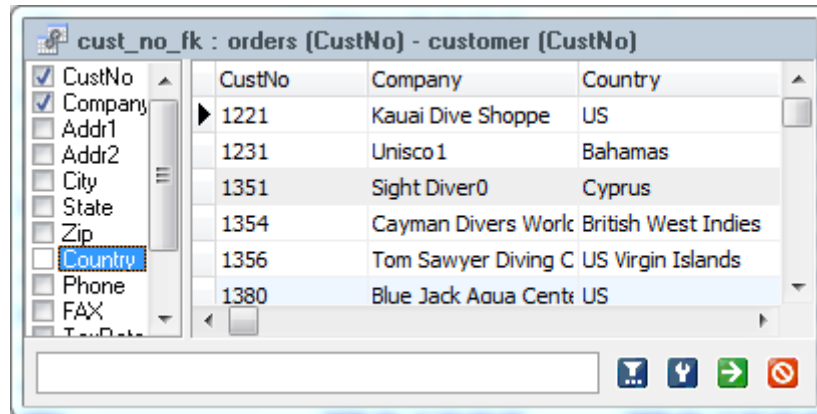
OrderNo	CustNo	SaleDate
1003	1351	2088-04-12 00:00:00
1004	2156	2088-04-17 00:00:00
▶ 1005	1356	 2088-04-20 00:00:00
1006	1380	2094-11-06 00:00:00

- Just simply double-click to select the desired data.



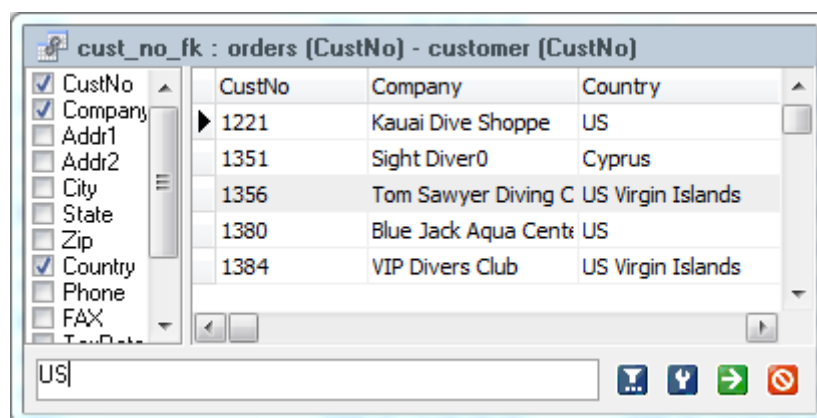
- Hint:**
1. By default, the number of records showed per page is **100**. To show all records, right-click anywhere on the grid and select **Show All**. To adjust the global settings, see Options.
  2. To refresh the record, right-click anywhere on the grid and select **Refresh** or press F5.

- Click  to open a panel on the left for showing a list of column name(s). Just simply click to show the additional column. To remove the column(s), uncheck them in the same way.



- Hint:**
- To set column in ascending or descending mode, right-click anywhere on the column and select **Sort** -> **Sort Ascending** / **Sort Descending**. Remember to remove all sorting before applying on another column.
  - To find for the text in the editor window, right-click anywhere on the grid and select **Find** or press Ctrl+F.

- Enter a value into the edit box and click  to filter for the particular record(s).



- Hint:** To remove the filter results, right-click anywhere on the grid and select **Show All**.

## Copying/Pasting Data from/into Navicat

Data that being copied from Navicat goes into the windows clipboard with the fields delimited by tabs and the records delimited by carriage returns. It allows you to easily paste the clipboard contents into any application you want. Spreadsheet applications in general will notice the tab character between the fields and will neatly separate the clipboard data into rows and columns.


### Copy Data from Navicat

To select data using **Keyboard Shortcuts**

Ctrl+A	Toggles the selection of all rows and columns in a data grid.
Shift+Up Arrow	Toggles the selection of rows as you move up in the data grid.
Shift+Down Arrow	Toggles the selection of rows in the data grid as you move down.

To select data using **Mouse Actions**

1. Highlighted the desired records by holding down the Ctrl key while clicking on each row.
2. Highlighted range of records by clicking the first row you want to select and holding down the Shift key together with moving your cursor to the last row you wish to select.


**Note:** After you have selected the desired records, just simply press Ctrl+C or right-click and select the  **Copy** from the popup menu.

### Paste Data into Navicat

Data is copied into the clipboard will be arranged as below format:

1. Data is arranged into rows and column.
2. Rows and columns are delimited by carriage returns/tab respectively.
3. Columns in the clipboard have the same sequence as the columns in the data grid you have selected.



When pasting data into Navicat, you can replace the contents of current records and append the clipboard data into the table. To replace the contents of current records in a table, one must select the rows in the data grid whose contents must be replaced by the data in the clipboard.

**Note:** Just simply press Ctrl+V or right-click and select the  **Paste** from the popup menu. The paste action cannot be undone.


## Sorting and Finding Records

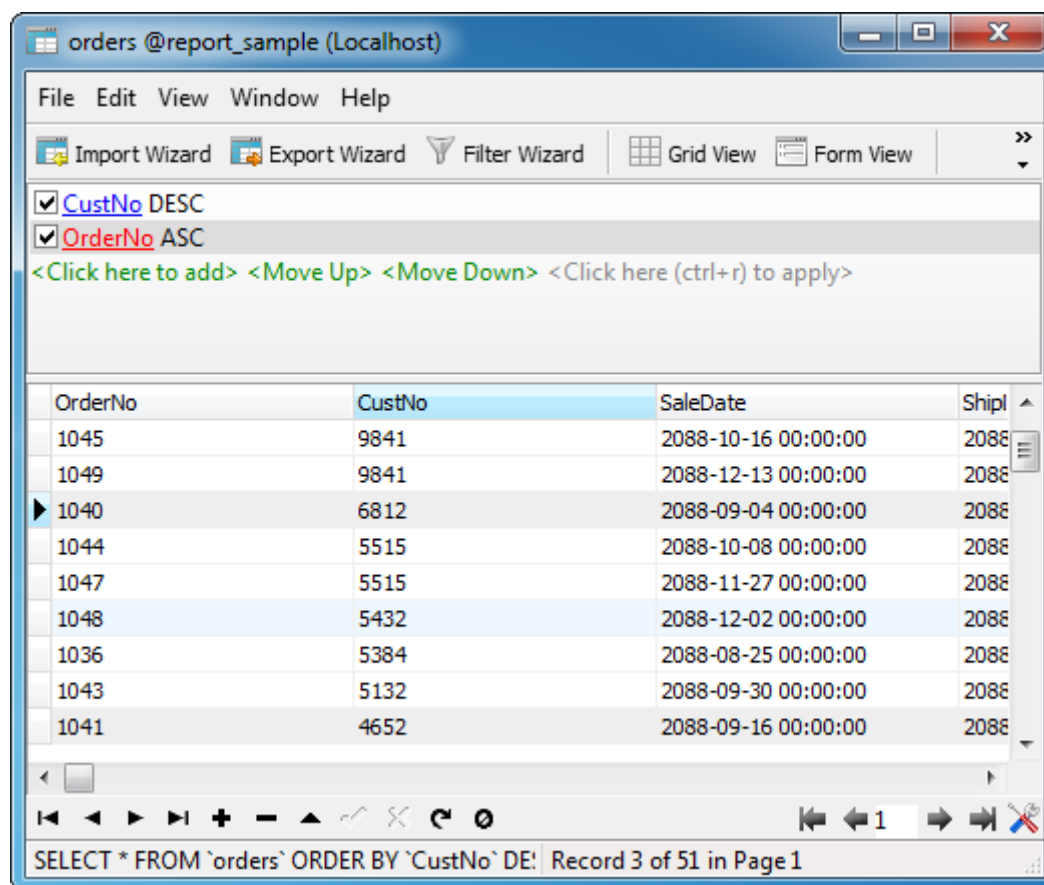
### Sorting Records

Server stores records in the order they were added to the table. Sorting in Navicat is used to temporarily rearrange records, so that you can view or update them in a different sequence.


Click the column caption whose contents you want to sort by, right-click to select the  **Sort Ascending** or  **Sort Descending** mode from the popup menu or choose from the toolbar.

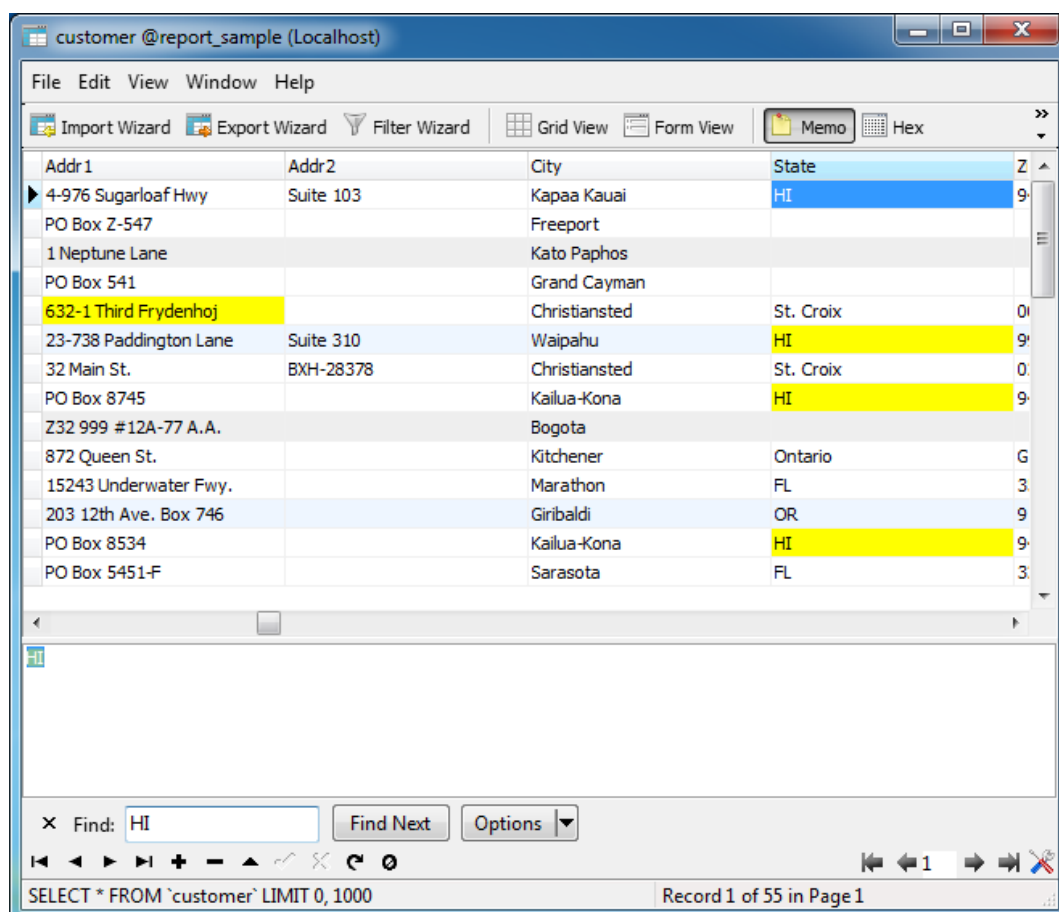
**Hint:** Remember to remove all sorting before applying on another column.

To sort by custom order of multi fields, right-click the grid to select the  **Custom Sort** mode from the popup menu or choose from the toolbar.



## Finding Records

The **Find** Dialog is provided for quick searching for the text in the editor window. Just simply click **Edit** ->  **Find** from the menu or press Ctrl+F and enter a search string.



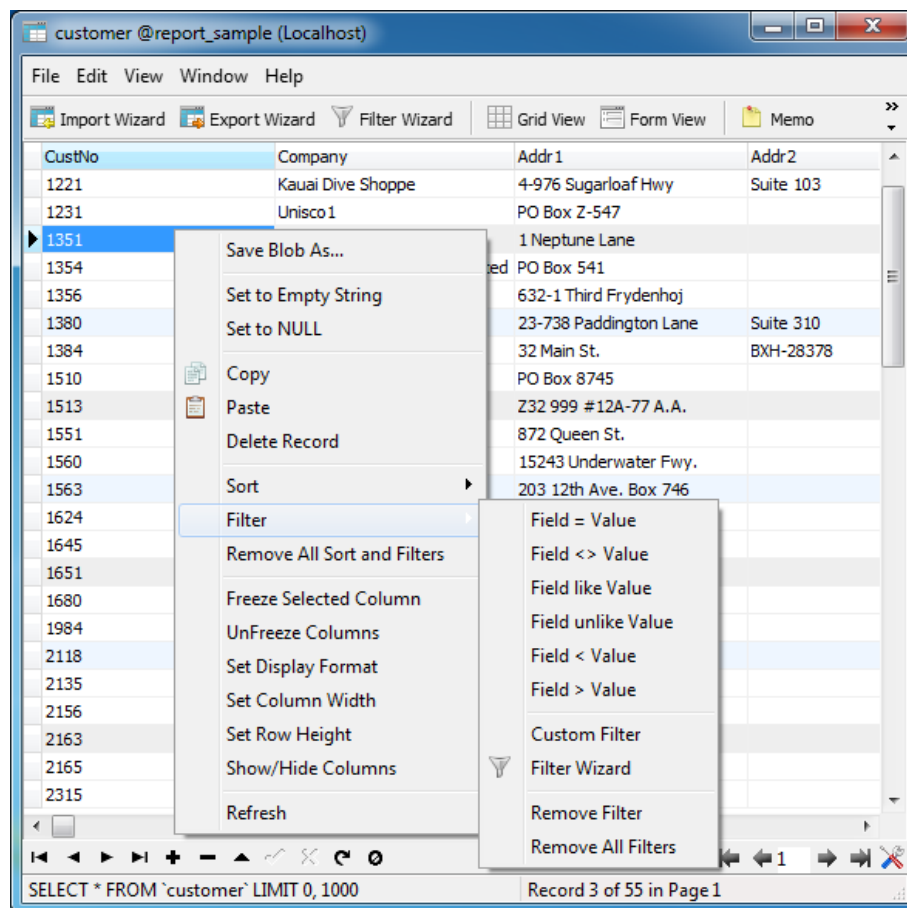
The search starts at the cursor's current position to the end of the file. There will not have differentiates when performing a uppercase or lowercase search.



To find for the next text, just simply select **Edit** -> **Find Next** or press F3.

## Filtering Records (Available only in Full Version)

Use either of the following methods to filter the data in the grid:

- Right-click a field and select the **Filter** from the popup menu to filter records by the current value of the selected column.



- The **Custom Filter** Dialog is provided for quick building a simple filter. Just simply right-click a field and select the **Filter** -> **Custom Filter** from the popup menu. Use character '\_' to represent any single symbol in the condition and use character '%' to represent any series of symbols in the condition.
- You can also customize your filter in a more complicated way by right-clicking a field and selecting the **Filter** ->  **Filter Wizard** from the popup menu or clicking the  **Filter Wizard** from the toolbar. The Filter Wizard becomes visible at the top of grid, where you can see the active filtering condition and easily enable or disable it by clicking a check box at the left.

## Manipulating Raw Data (Available only for MySQL, PostgreSQL, SQLite and SQL Server)

Navicat normally recognize what user has input in grid as normal string, any special characters or functions would be processed as plain text (that is, its functionality would be skipped).

Editing data in **Raw Mode** provides an ease and direct method to apply server built-in function. To access the Raw Mode function, just simply select **View** -> **Raw Mode** from the menu.

CustNo	Company	Addr1
'1384'	'VIP Divers Club'	'32 Main St.'
'1510'	'Ocean Paradise'	'PO Box 8745'
I '1513'	Concat('Fantastique', ' Aquatica')	'232 999 #12A-77 A.A.'
'1551'	'Marmot Divers Club'	'872 Queen St.'
'1560'	'The Depth Charge'	'15243 Underwater Fwy.'



## Formatting Table Grid

Use the following methods to format the table grid:

### Move Columns

1. Click on the column header and hold down the left mouse button.
2. Move the pointer until a double black line appears in the desired location.
3. Release the mouse and the column will move.

CustNo	Company	Addr1
1221	Kauai Dive Shoppe	4-976 Sugarloaf Hwy
1231	Unisco1	PO Box Z-547
1351	Sight Diver0	1 Neptune Lane
1354	Cayman Divers World Unlimited	PO Box 541
1356	Tom Sawyer Diving Centre	632-1 Third Frydenhoj
1380	Blue Jack Aqua Center	23-738 Paddington Lane
1384	VIP Divers Club	32 Main St.
1510	Ocean Paradise	PO Box 8745
1513	Fantastique Aquatica	Z32 999 #12A-77 A.A.

### Freeze Selected Column

If there are many columns in the table and you want to freeze one or more columns to identify the record. Just simply right-click the column you want to freeze and select **Freeze Selected Column** or select **View -> Freeze Selected Column** from the menu.

The frozen column(s) will move to the leftmost position in the table grid. This action will locks the frozen column(s), preventing them from being edited.

To unfreeze the columns, just simply right-click anywhere on the table grid and select **Unfreeze Columns** or select **View -> Unfreeze Columns** from the menu.

### Set Display Format

The **Set Display Format** Dialog is provided for you to customize format applied to exported data on the selected column. Just simply right-click the column you want to edit its format and select **Set Display Format...** or select **View -> Set Display Format...** from the menu. Edit the format style to adjust the result format in the way you need. For example: dd-mm-yyyy.

**Hint:** This action applies on the selected column only. To adjust the global settings, see Options.

## Set Column Width

Click right border at top of column and drag either left or right.

or

Double-click right border at top of column to obtain the best fit for the column.

or

Right-click the column you want to set the column width with and select **Set Column Width...** or select **View -> Set Column Width...** from the menu. Specify width in the **Set Column Width** Dialog. The default value is 120.

**Hint:** The result only applies on the selected column. To adjust the global settings, see Options.

## Set Row Height

Right-click anywhere on the table grid and select **Set Row Height...** or select **View -> Set Row Height...** from the menu. Specify row height in the **Set Row Height** Dialog. The default value is 17.

**Hint:** This action applies on the current table grid only. To adjust the global settings, see Options.

## Show/Hide Columns

If there are many columns in the table and you want to hide some of them from the table grid. Just simply right-click anywhere on the table grid and select **Show/Hide Columns** or select **View -> Show/Hide Columns** from the menu. Select the columns that you would like to hide.

The hidden column(s) will disappear from the table grid.

To unhide the columns, just simply right-click anywhere on the table grid and select **Show/Hide Columns** or select **View -> Show/Hide Columns** from the menu. Select the columns that you would like to redisplay.

<input checked="" type="checkbox"/> CustNo	CustNo	Addr1
<input type="checkbox"/> Company	1221	4-976 Sugarloaf Hwy
<input checked="" type="checkbox"/> Addr1	1231	PO Box Z-547
<input checked="" type="checkbox"/> Addr2	1351	1 Neptune Lane
<input type="checkbox"/> City	1354	PO Box 541
<input type="checkbox"/> State	1356	632-1 Third Frydenhoj
<input checked="" type="checkbox"/> Zip	1380	23-738 Paddington Lane
<input checked="" type="checkbox"/> Country		

## Show/Hide ROWID (Available only for Oracle and SQLite)

If you want to display or hide the rowid (address) of every row, right-click anywhere on the table grid and select **Show/Hide ROWID** or select **View -> Show/Hide ROWID** from the menu.

The column **ROWID** will be showed in the last column.

	MIN_SALARY	MAX_SALARY	ROWID
	20000	40000	AAARG3AAFAAAAC9AAA
	15000	30000	AAARG3AAFAAAAC9AAB
▶	3000	6000	AAARG3AAFAAAAC9AAC
	8200	16000	AAARG3AAFAAAAC9AAD
	4200	9000	AAARG3AAFAAAAC9AAE
	8200	16000	AAARG3AAFAAAAC9AAF

## Form View (Available only in Full Version)

The form view allows you to view, update, insert, or delete data as a form, which the current record is displayed: field name and its value. The popup menu of the form provides the following additional functions: set the field value as Null/Empty String, use current field value as a filter, format form view, and more.

The navigation bar allows you to switch the records quickly, insert, update or delete records.

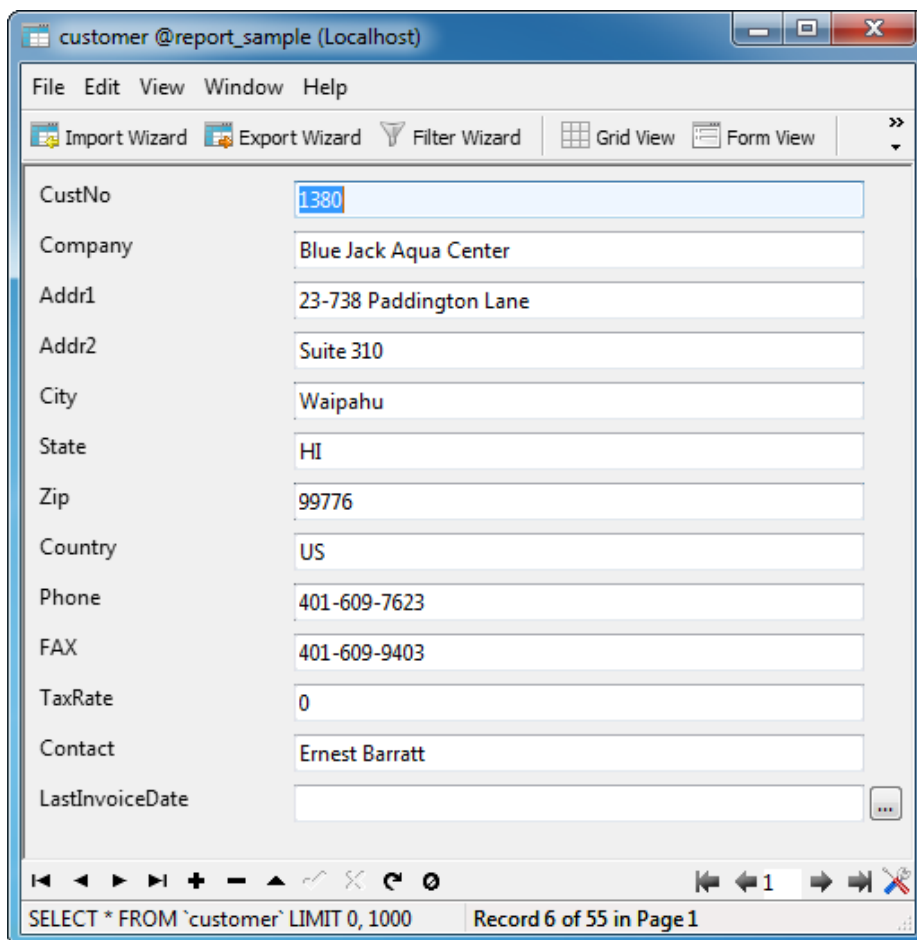
Related topic:

[Sorting and Finding Records](#)

[Filtering Records](#)

[Manipulating Raw Data](#)

[Formatting Table Grid](#)



The screenshot shows the Navicat Form View for a table named 'customer' on a 'report\_sample' database at 'localhost'. The window has a menu bar (File, Edit, View, Window, Help) and a toolbar with icons for Import Wizard, Export Wizard, Filter Wizard, Grid View, and Form View. The form displays the following fields and values:




CustNo	1380
Company	Blue Jack Aqua Center
Addr1	23-738 Paddington Lane
Addr2	Suite 310
City	Waipahu
State	HI
Zip	99776
Country	US
Phone	401-609-7623
FAX	401-609-9403
TaxRate	0
Contact	Ernest Barratt
LastInvoiceDate	

At the bottom, there is a navigation bar with icons for navigating between records. Below the navigation bar, the SQL query 'SELECT \* FROM `customer` LIMIT 0, 1000' is displayed, along with the status 'Record 6 of 55 in Page 1'.

## Text/Blob/BFile View

Navicat provides Text/Blob/BFile Viewer and Editor to view and edit TEXT/BLOB/BFile fields content. The editor allows you to view, update, insert, or delete data in a table.

**Note:** Oracle BFile fields cannot be edited.

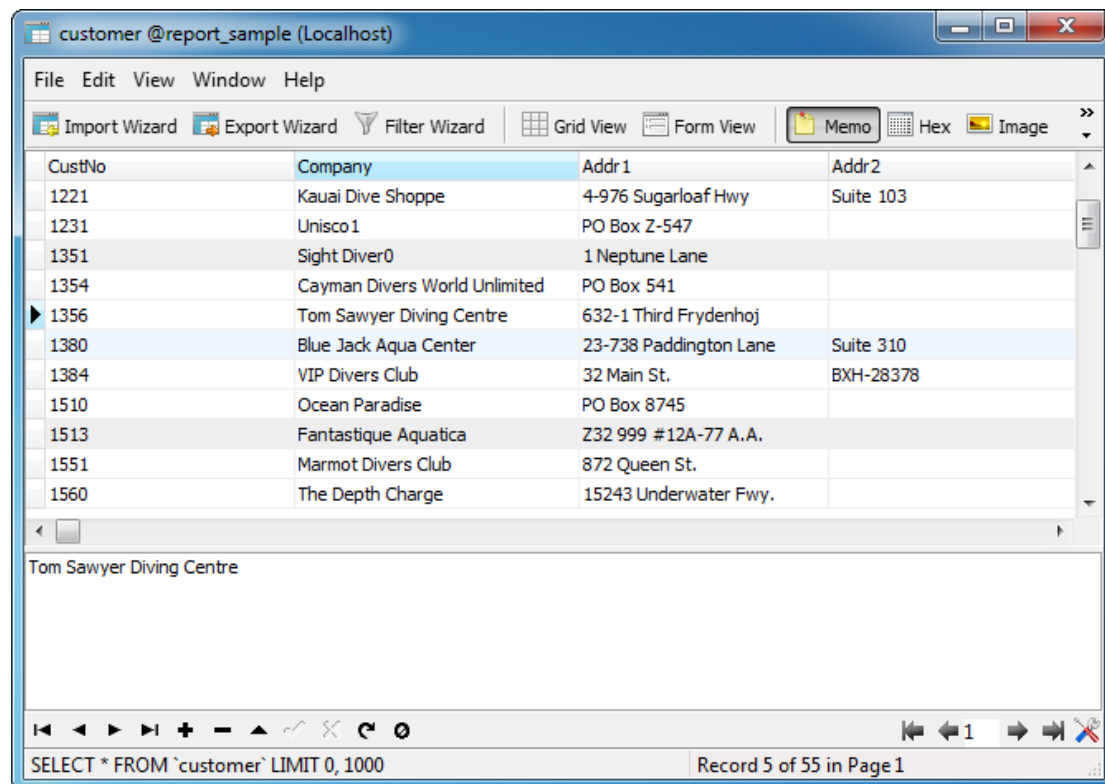
Click  **Memo**,  **Hex** and  **Image** from the toolbar to activate the appropriate viewer/editor.

- [Viewing/Editing Text/BFile field as Memo](#)
- [Viewing/Editing Text/Blob/BFile field as Hexadecimal](#)
- [Viewing/Editing Blob/BFile field as Graphical Image](#)

## Viewing/Editing Text/BFile field as Memo

The **Memo** panel allows you to edit data as a simple text. Use the ☒ button on the navigation bar to update the changed records to the table.

**Note:** Oracle BFile fields cannot be edited.

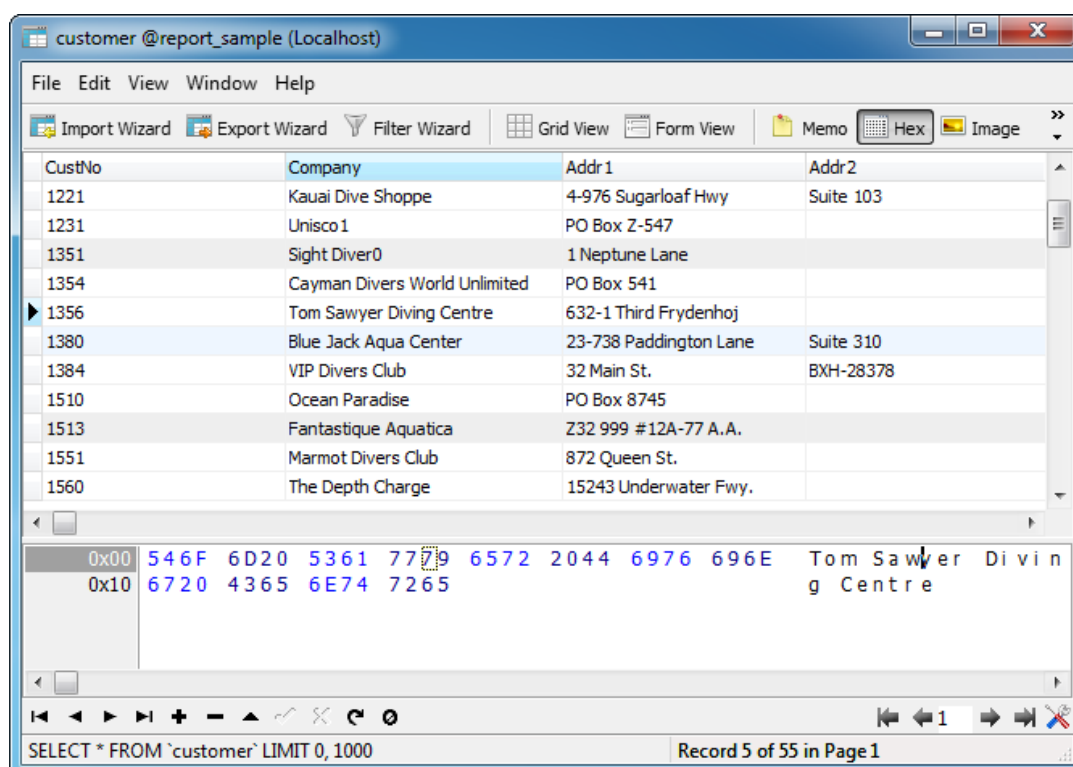


## Viewing/Editing Text/Blob/BFile field as Hexadecimal




The **Hex** panel allows you to edit data in hexadecimal mode. Use the ☒ button on the navigation bar to update the changed records to the table.

**Note:** Use the **Insert** key on the keyboard to switch between Insert and Overwrite modes

**Note:** Oracle BFile fields cannot be edited.

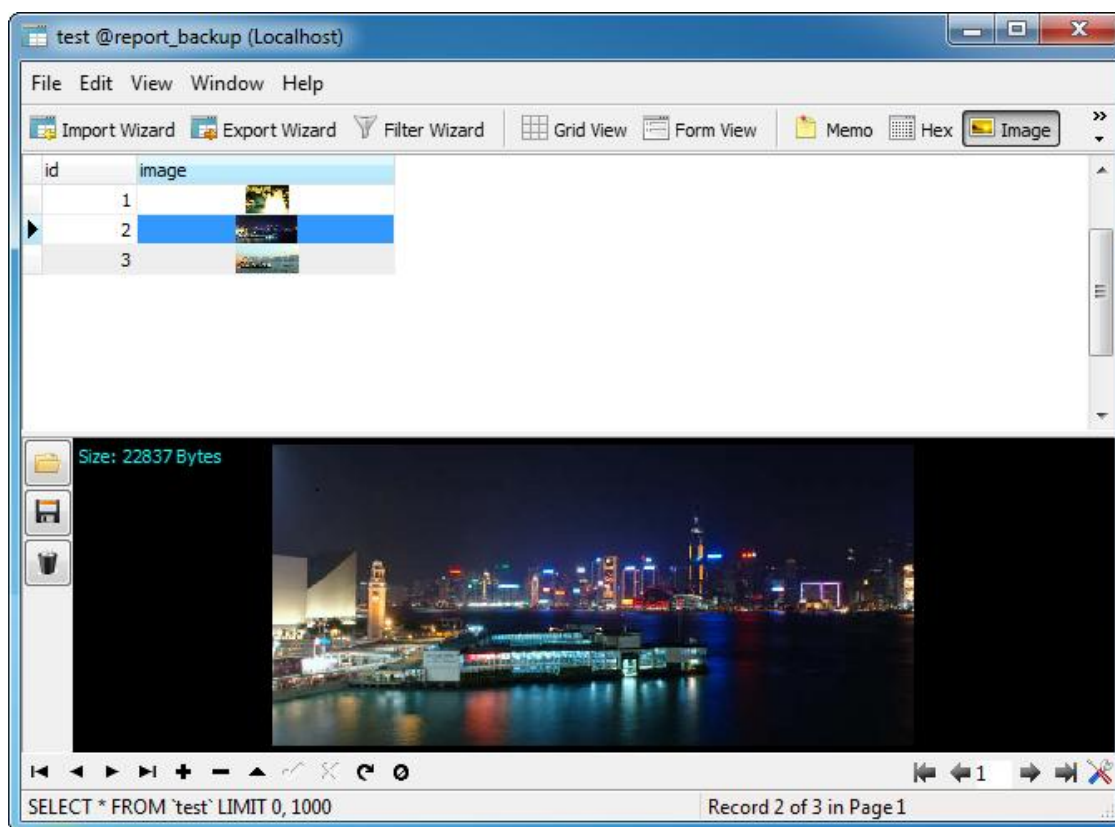


## Viewing/Editing Blob/BFile field as Graphical Image

The **Image** panel allows you to show data as image. Use the  **Load**,  **Save to disk** and  **Clear** button to load/remove the image from a file, or save the image to a table.

**Note:** Oracle BFile fields cannot be edited.

You can also right-click a Blob field type and save to disk.






## Filter Wizard (Available only in Full Version)

Filter Wizard allows you to facilitate creating and applying filter criteria that you specify for the table grid. Moreover, it allows you to save filter criteria as a profile for future use.

Click  **Filter Wizard** from the toolbar to activate the editor.

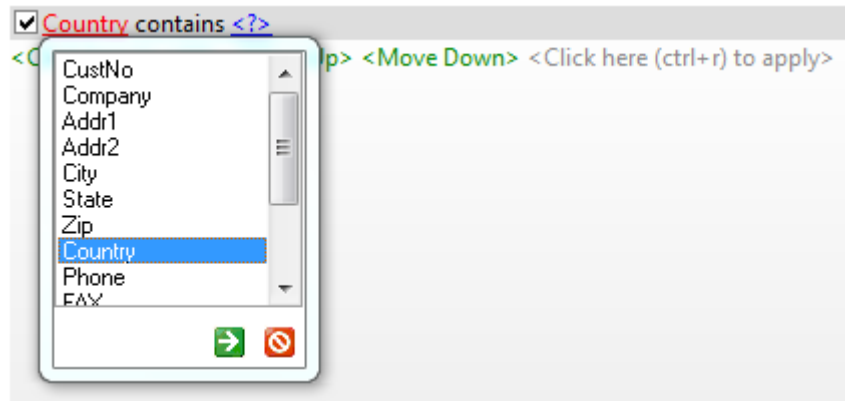
- [Adding New Filter Condition](#)
- [Setting Filter Criteria](#)
- [Setting Filter Operator](#)
- [Setting Filter Criteria Values](#)
- [Setting Filter Group](#)
- [Applying Filter Conditions](#)

## Adding New Filter Condition

To add a new condition to the criteria, just simply click the **<Click here to add>** or right-click anywhere on the Filter Wizard and select the  **Add** from the popup menu.

## Setting Filter Criteria

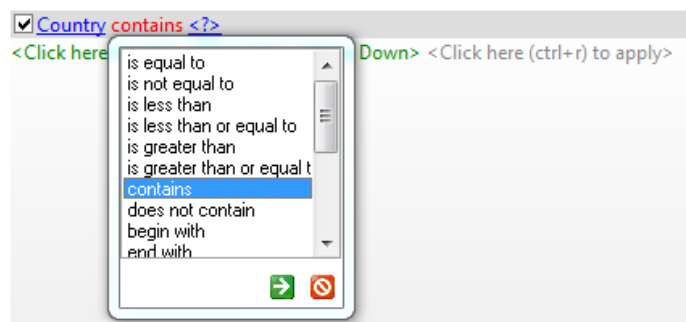
Suppose we need to select customers who come from **US**. This criteria is applied to the **Country** column. Click on the column box (next to the check box) and select **Country** item from the dropdown list which displaying all available column names.



## Setting Filter Operator

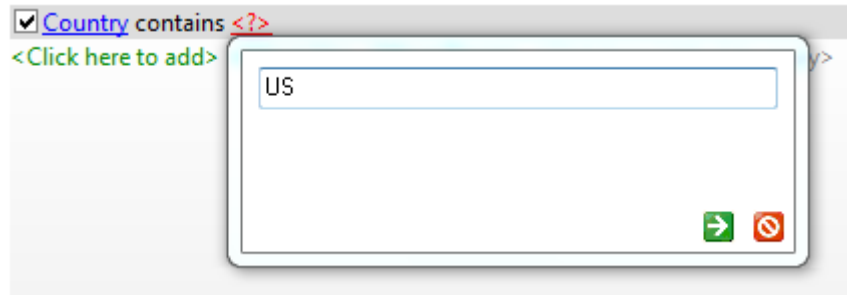
Click on the filter operator box (next to the column box) and select **contains** item from the dropdown list which displaying all available filter operators.

Filter Operator	Result
is equal to <?>	My_Field = 'your_value'
is not equal to <?>	My_Field <> 'your_value'
is less than <?>	My_Field < 'your_value'
is less than or equal to <?>	My_Field <= 'your_value'
is greater than <?>	My_Field > 'your_value'
is greater than or equal to <?>	My_Field >= 'your_value'
contain <?>	My_Field LIKE '%your_value%'
does not contain <?>	NOT (My_Field LIKE '%your_value%')
begin with <?>	My_Field LIKE 'your_value%'
end with <?>	My_Field LIKE '%your_value'
is null	My_Field IS NULL
is not null	My_Field IS NOT NULL
is empty	My_Field = ''
is not empty	My_Field <> ''
is between <?> <?>	((My_Field >= your_value1) and (My_Field <= your_value2))
is not between <?> <?>	NOT ((My_Field >= your_value1) and (My_Field <= your_value2))
is in list <?>	My_Field in ('aaa','bbb',...)
is not in list <?>	My_Field not in ('aaa','bbb',...)



## Setting Filter Criteria Values

Click on the criteria values box (next to the filter operator box) to activate the appropriate editor and enter the criteria values **(US)**. The editor used in criteria value boxes is determined by the editor type assigned to the corresponding column.

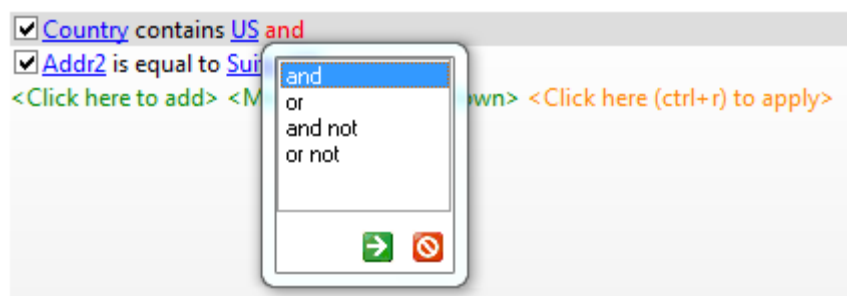


## Setting Filter Group

To implement a complex filter condition combining two simple conditions, just simply click on the **<Click here to add>** under the existing condition you have just defined.

Suppose we need to select customers whose **Company** located in **Suite 103, US**. Apply **Addr2** on column box, **is equal to** on filter operator box and **Suite 103** on criteria values box under the existing condition.

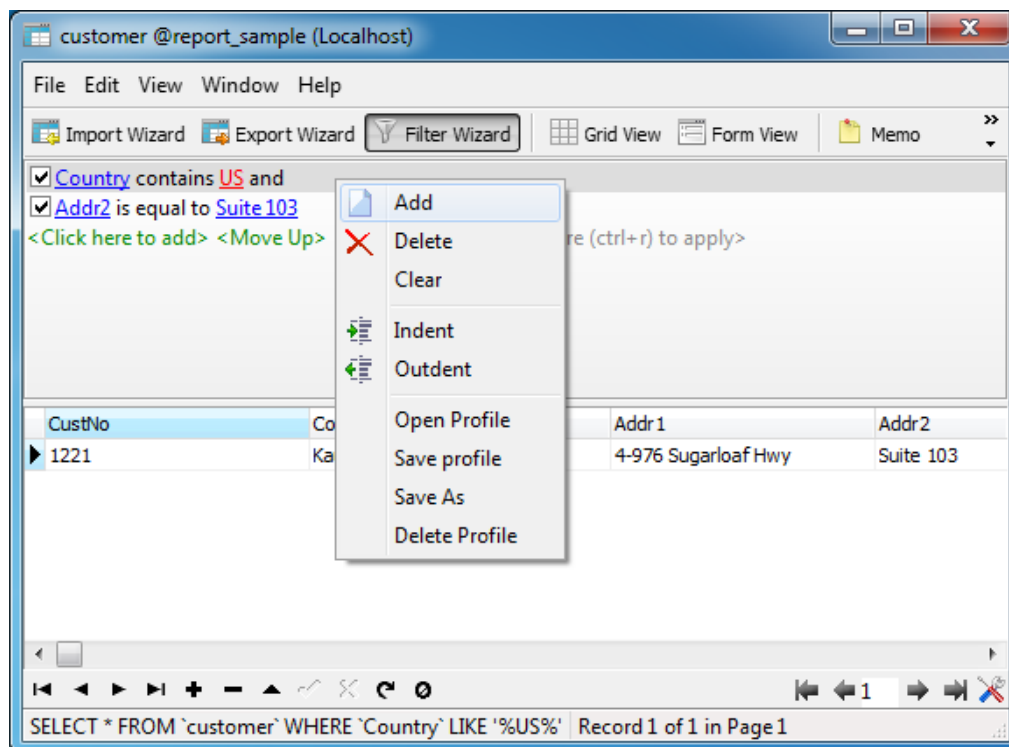
Setting filter group operator for the combine conditions, just simply click on the filter group operator box (by default, it specifies **AND** operator) to activate the appropriate editor.



## Applying Filter Conditions


Click the **<Click here (ctrl+r) to apply>** or press **Ctrl+r** to see the result of the filtering you made.

**Hint:** You are allowed to save filter criteria to and load them from the registry for future use. Just simply right-click on the Filter Wizard and select **Save Profile / Open Profile**.





## Queries

A query is used to extract data from the database in a readable format according to the user's request. Navicat provides two powerful tools for working with the SQL queries: Query Editor for editing the query text directly and Query Builder for building queries visually. You can save your queries for setting schedule.



Just simply click  to open an object pane for **Query**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit, open and delete the selected query.

### Create Query




To create a new query in Query Editor


- Select anywhere on the object pane.
- Click the  **New Query** from the object pane toolbar.  
or
- Right-click and select  **New Query** from the popup menu.
- Edit query text on the Query Editor tab.

To create a new query in Query Builder

- Select anywhere on the object pane.
- Click the  **New Query** from the object pane toolbar.  
or
- Right-click and select  **New Query** from the popup menu.
- Edit query on the Query Builder tab.

To create a new query with loading from a SQL file

- Select anywhere on the object pane.
- Click the  **New Query** from the object pane toolbar.  
or
- Right-click and select  **New Query** from the popup menu.
- Click  **Load**.

**Hint:** To create new query you can also right-click the Queries node of the navigation pane and select the  **New Query** from the popup menu.



To create a new query with the same properties as one of the existing queries has (using drag and drop method)




**Apply to:** current database {same connection}

- Select the query(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen query(s) to the target location.
- Select one of the following options:
  - Copy here
  - Move here
  - Cancel
- The newly created query(s) will be named as "queryname\_**copy**"

**Apply to:** different database {same connection}  
different database {different connection}

- Select the query(s) for copying in the object pane.
- Drag and drop the chosen query(s) to the target database.
- Select one of the following options:
  - Copy here
  - Move here
  - Cancel

To create a new query with modification as one of the existing queries



- Select the query for modifying in the navigation pane/object pane.
- Right-click and select the  **Design Query** from the popup menu.  
or
- Click the  **Design Query** from the object pane toolbar.
- Modify query on the Query Editor/Query Builder tab.
- Click  **Save As**.

**Hint:** Queries(.sql) are saved under the Settings Save Path.



## Edit Query

To edit the existing query



- Select the query for editing in the navigation pane/object pane.
- Right-click and select the  **Design Query** from the popup menu.  
or
- Click the  **Design Query** from the object pane toolbar.
- Modify query on the Query Editor/Query Builder tab.

To change the name of the query

- Select the query for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.

## Open Query



To open a query (manage query data)

- Select the query for opening in the navigation pane/object pane.
- Right-click and select the  **Open Query** from the popup menu or simply double-click the query.  
or
- Click the  **Open Query** from the object pane toolbar.

**Note:** Only *SELECT* queries will be run automatically with results being displayed on Result tab.

## Run Query

To run a query



- Create a new query/open the existing query.
- Click  **Run**. (Click  **Stop** to terminate the running)
- View/edit the returned data on the Result tab.

To run a saved query from the command line (set schedule)

- Create and save the query.
- In terminal, type the command (see Command for details)

## Delete Query

To delete a query

- Select the query for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Query** from the popup menu.  
or
- Click the  **Delete Query** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Query Information

To achieve a query information

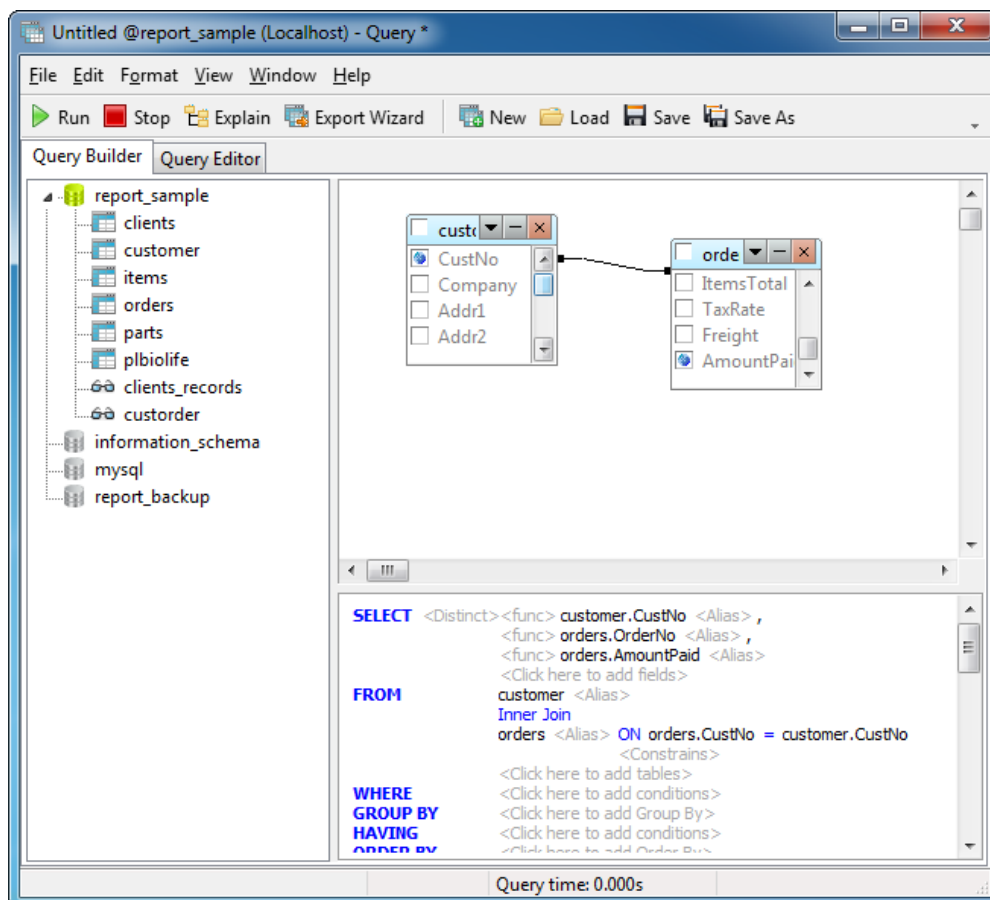
- Select the query in the navigation pane/object pane.
- Right-click the selected query and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Query Builder (Available only in Full Version)

Navicat provides a useful tool called **Visual Builder** for building queries visually. It allows you to create and edit queries without knowledge of SQL. The database objects are displayed in left panel. Whereas in the right panel, it is divided into two portions: the upper **Graphical View**, and the lower **Syntax View**.

**Note:** Visual Builder supports *SELECT* statement only. Use Query Editor for creating complex queries.

- [Working with Diagram Area](#)
- [Setting Field Association](#)
- [Setting Output Fields](#)
- [Setting Criteria](#)
- [Setting Grouping Criteria](#)
- [Setting Sorting Criteria](#)
- [Setting Limit Criteria](#) (Available only for MySQL, PostgreSQL and SQLite)



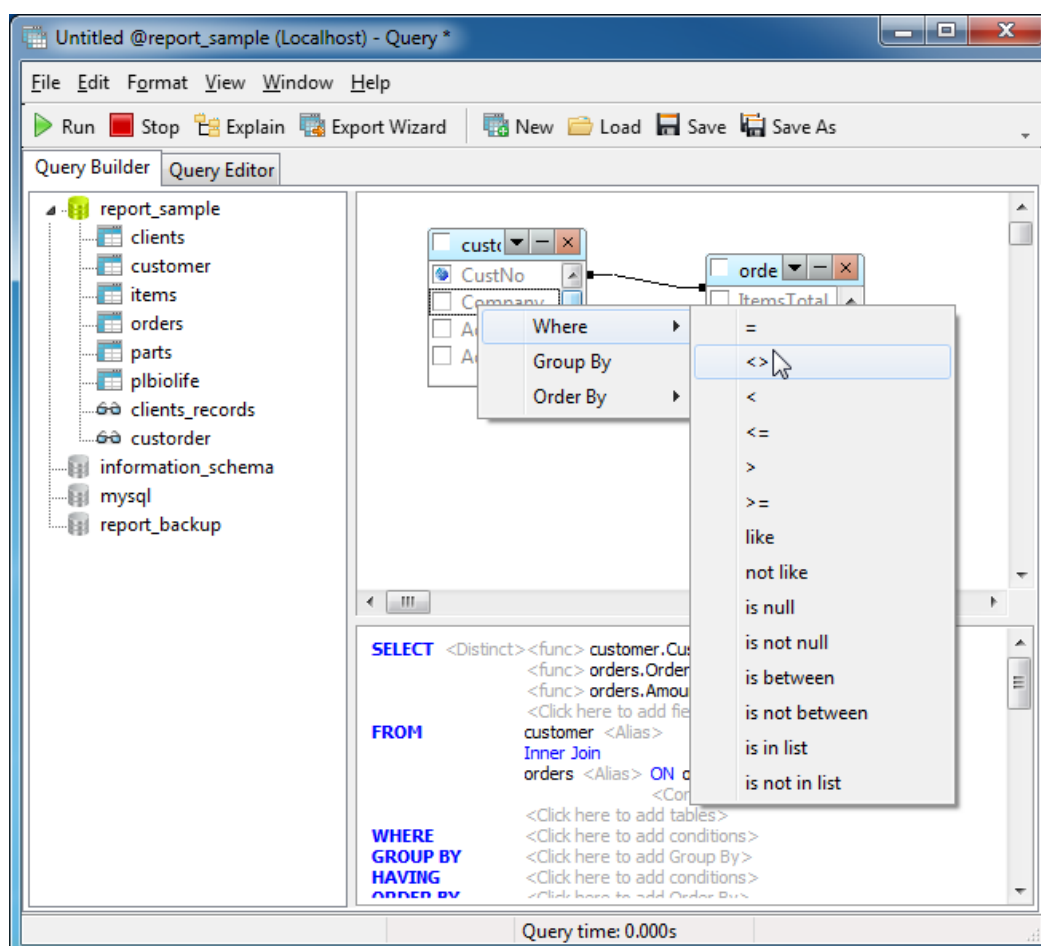
## Working with Diagram Area

To add a table to the query, simply drag it or double-click from the left panel to the Graphical View area. To include a table field in the query, check the left of the field name in the list. To include all the fields, click at the left of the table caption.

To remove the object from the Graphical View area, click the cross button at the object caption.

To add the table alias, simply double-click the table name and enter the alias in the the Graphical View area.

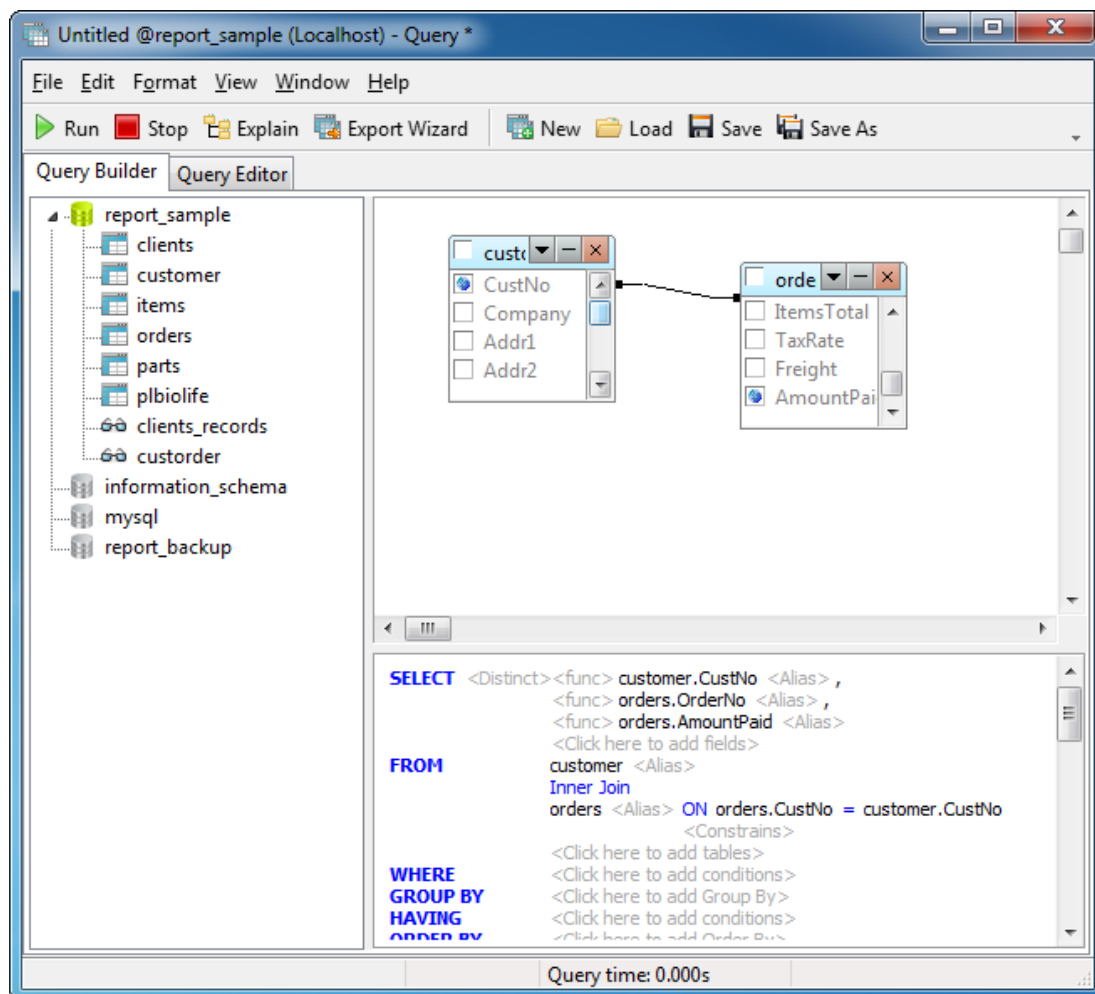
**Hint:** You are also allowed to set criteria by right-clicking any fields from the Graphical View area.



## Setting Field Association

To associate database objects by two fields, just drag one field from the object list to another and a line will appear between the linked fields.

**Hint:** To delete all the links of some object, click button '-' next to the object alias.



## Setting Output Fields

The fields you have selected in the graphical view will be displayed in the Syntax View which allows you to set their displaying order and modify the output fields of the query using **<Distinct>**, **<func>** and **<Alias>**.

### <Distinct>

Enable this option if you wish the repeated records are not included into the query result.

### <func>

Set the aggregate functions (SUM, MAN, MIX, AVG, COUNT) for each field.

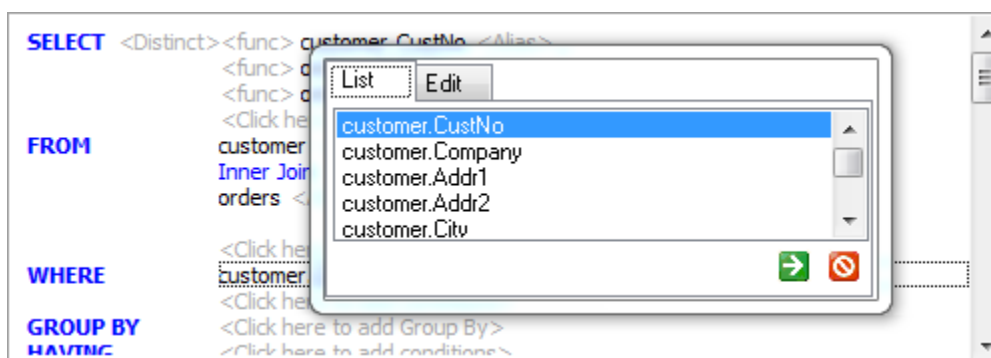
### <Alias>

Change the output query field name.

```
SELECT <Distinct><func> customer.CustNo <Alias> ,
      <func> orders.OrderNo <Alias> ,
      Sum(orders.AmountPaid) <Alias>
      <Click here to add fields>
FROM   customer <Alias>
      Inner Join
      orders <Alias> ON orders.CustNo = customer.CustNo
      <Constrains>
      <Click here to add tables>
WHERE  <Click here to add conditions>
GROUP BY customer.CustNo,
          orders.OrderNo
      <Click here to add Group By>
```

## Setting Criteria

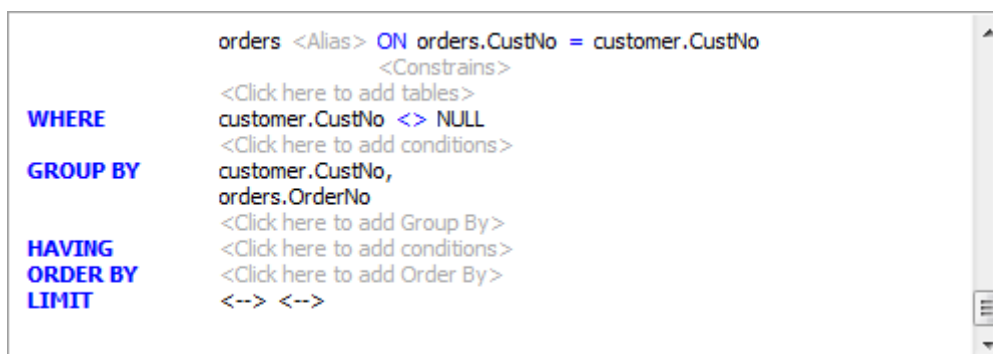
To add a condition, click the **<--> = <-->** from the **Where** Clause in the Syntax View. Click **<-->** to choose the field from the list of all the table fields, available in the query. To define your own criteria, type your values directly in the Edit Tab. Clicking **=** to set condition operator.





## Setting Grouping Criteria

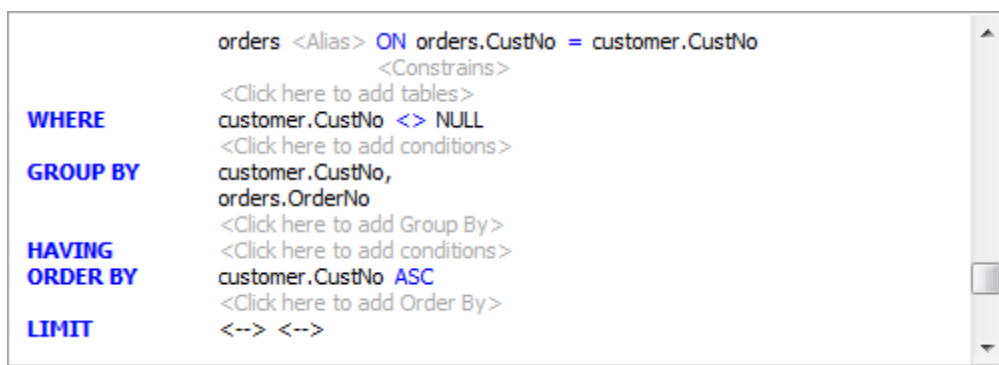
You can set the conditions for grouping query records from the **Group By** Clause in the Syntax View. They are set in the same way as setting criteria. The conditions will be included into the **HAVING** statement of the current query.



## Setting Sorting Criteria

When you query your database/schema, you can sort the results by any field in an ascending or descending order by just adding *ORDER BY* at the end of your query.

In Visual Builder, you can set the way of sorting query records from the **Order By** Clause in the Syntax View. To change the sorting direction, click on either **ASC** or **DESC**.



## Setting Limit Criteria (Available only for MySQL, PostgreSQL and SQLite)

**Limit** Clause is used to limit your query results to those that fall within a specified range. You can use it to show the first X number of results, or to show a range from X - Y results. It is phrased as Limit X, Y and included at the end of your query. X is the starting point (remember the first record is 0) and Y is the duration (how many records to display).

	orders <Alias> ON orders.CustNo = customer.CustNo
	<Constrains>
	<Click here to add tables>
WHERE	customer.CustNo <> NULL
	<Click here to add conditions>
GROUP BY	customer.CustNo,
	orders.OrderNo
	<Click here to add Group By>
HAVING	<Click here to add conditions>
ORDER BY	customer.CustNo ASC
	<Click here to add Order By>
LIMIT	1 , 6

## Query Editor

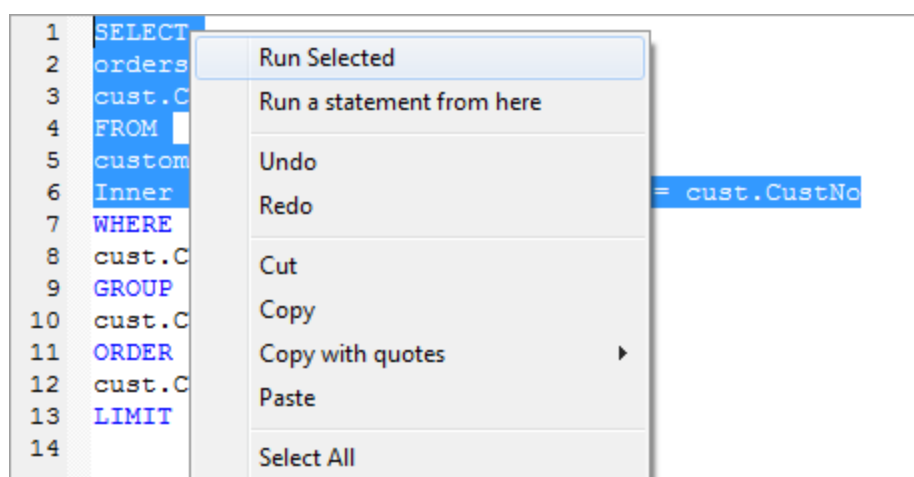
Navicat provides a useful tool called **Query Editor** for creating and executing queries. It allows you to create and edit SQL text for a query, prepare and execute selected queries.

You can show the object tree, simply choose View -> **Show Object Tree**.

**Hint:** Query text will be automatically generated while you build in Query Builder.

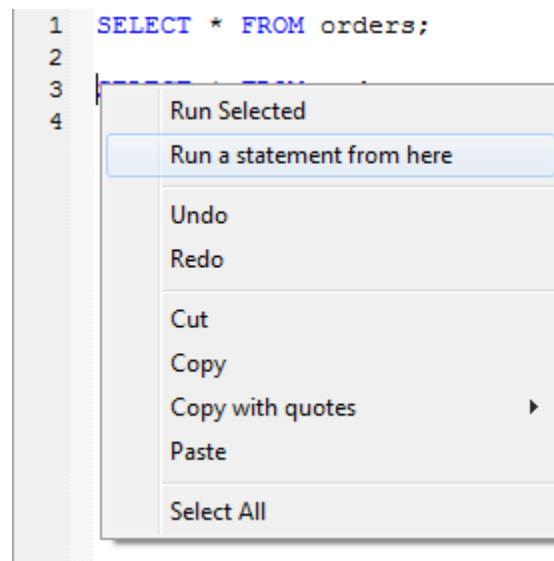
Query Builder	Query Editor
1	SELECT
2	orders.EmpNo,
3	cust.CustNo
4	FROM
5	customer AS cust
6	Inner Join orders ON orders.CustNo = cust.CustNo
7	WHERE
8	cust.CustNo > 1000
9	GROUP BY
10	cust.CustNo
11	ORDER BY
12	cust.CustNo ASC
13	LIMIT 1, 5
14	

You are allowed to run selected portion of query, just simply right-click the highlighted query and select **Run Selected**.



You can define multiple SQL statements in one Editor window, and the editor let you run the current statement your cursor is on (place your cursor in the front of the desired statement). Just simply select **Run a statement from here** or press **F7**.

**Note:** Select **Run a statement from here** or press **F7**, the next statement will be continue to run.



## Editor View and More Features

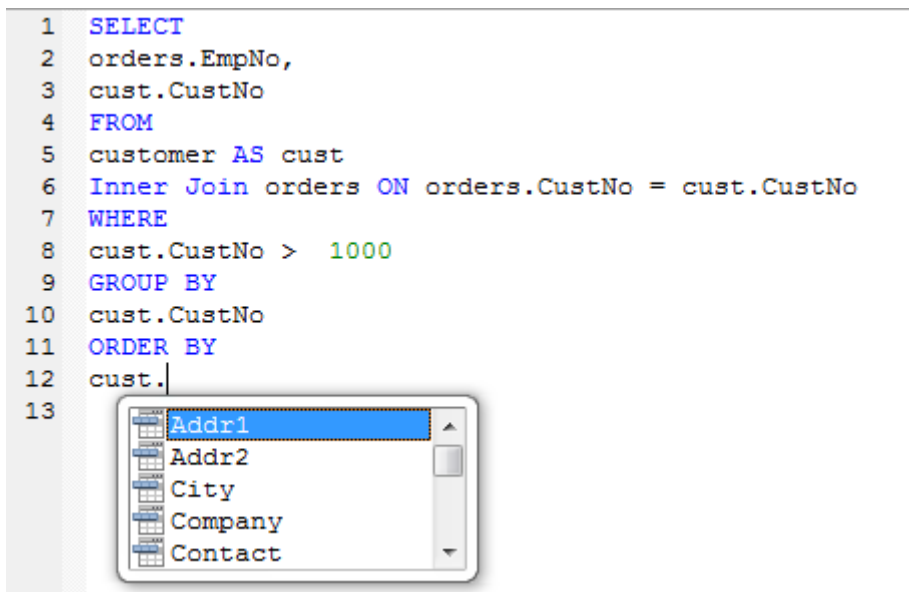
Navicat allows you to customize the view of the **SQL Editor** and provides a wide range of compelling code editing capabilities, smart code-completion, sql formatting, and more.

- [Code-Completion](#)
- [SQL Formatting](#)
- [Code Folding](#)
- [Brace Highlight](#)
- [Find and Replace](#)
- [Copy With Quotes](#)
- [Zoom In/Zoom Out](#)

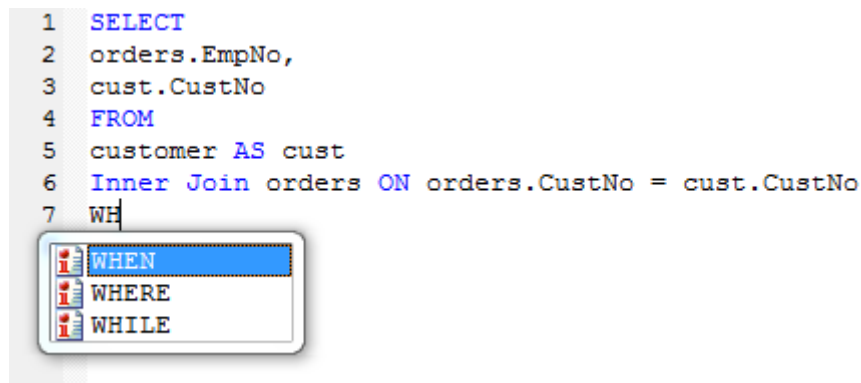
## Code-Completion (Available only in Full Version)

Code-completion in Navicat displays information in drop-down lists as you type your SQL statement in the editor, it assists you with statement completion and the available properties of database objects, for example databases, tables, fields, views etc with their appropriate icons.

To activate the code-completion, just simply press '.' for the available properties of database object currently in the scope.



**Hint:** You may invoke code-completion by typing two characters or pressing Ctrl+SPACE on your keyboard for SQL keywords.



**Hint:** Smart code-completion will pop-up a list of variants for the word completion automatically.

**Note:** Code-completion can be also applied on View, Functions/Procedures, etc.



## SQL Formatting (Available only in Full Version)

To change the SQL statement format, simply choose from the **Edit** menu -

### Indent

Increase/decrease indent for the selected lines of codes.

### Comment

Comment/uncomment the selected lines of codes.

### Convert Case

Format the selected codes into upper/lower case.

### Beautify SQL

Format the selected codes with the Beautify SQL settings.

### Beautify SQL With...

Change the Beautify SQL settings.

#### ☒ **Use tab character**

Check this option to use tab character.

#### **Tab size**

Set the tab size.

#### **Short brace length**

Set the length of the short brace.

#### ☒ **Upper case keywords**

Format all the SQL keywords to upper case.

#### ☒ **Save settings**





Save the SQL beautify options settings after you click **Beautify** button.

### Minify SQL






Minify the format of the SQL in the SQL Editor.

## Code Folding

Code folding feature enables you to collapse blocks of code such that only the first line of the block appears in **Editor**.

A block of code that can be folded is indicated by an icon  to the left of the first line of the block. A vertical line extends from the icon to the bottom of the foldable code. In contrast, a folded block of code is indicated by an icon  to left of the code block. You can fold the block by clicking  or expand it by clicking  in **Editor**.

```

1  CREATE OR REPLACE
2   PROCEDURE SETEMVIEWUSERCONTEXT(...)
7  IS
8   BEGIN
9
10  IF...END IF;
13
14  MGMT_VIEW_PRIV.SET_VIEW_USER_CONTEXT(
    em_client_in, op_in, view_user, view_pwd);
15
16  IF time_coff IS NOT NULL THEN
17     MGMT_TIME_SYNC.setTimeCoffNoCheck(null);
18 END IF;
19
20 EXCEPTION
21     WHEN OTHERS THEN
22  IF time_coff IS NOT NULL THEN
23     MGMT_TIME_SYNC.setTimeCoffNoCheck(null
24 );
25     END IF;
26     RAISE;
27 END;
```

## Brace Highlight

Navicat supports to highlight the matching brace in the editor, i.e. (), Begin...End .

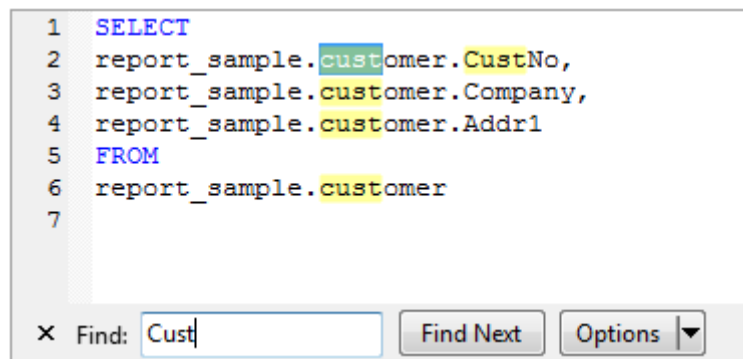
**Note:** The cursor must be on a brace to show the highlight.

```
1 CREATE OR REPLACE
2 PROCEDURE SETEMVIEWUSERCONTEXT(em_client_in
3   IN VARCHAR2,
4     op_in      IN INTEGER,
5     view_user  IN VARCHAR2,
6     view_pwd   IN VARCHAR2,
7     time_coff  IN VARCHAR2 DEFAULT NULL)
8 IS
9 BEGIN...END;
```

## Find and Replace

### Find

The **Find** Dialog is provided for quick searching for the text in the editor window. Just simply click **Edit** -> **Find** from the menu or press Ctrl+F and enter a search string.



The search starts at the cursor's current position to the end of the file. There will not have differentiates when performing a uppercase or lowercase search.

To find for the next text, just simply select **Edit** -> **Find Next** or press F3.

### Replace

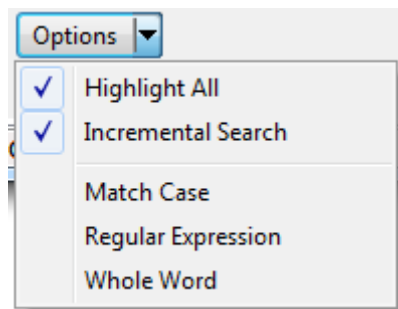
To open the **Replace** Dialog, simply click **Edit** -> **Replace** from the menu and enter the text you want to search and replace.

Click **Replace** button to replace the first occurrence.

Click **Replace All** button to replace all occurrences automatically .



There are some additional **Options** for Find and Replace:



## Copy With Quotes

To copy the SQL statement with quotes, just simply right-click the highlighted SQL. Then, select **Copy with quotes** and choose the format.

**Note:** Only available in Query, View and Materialized View.

## Zoom In/Zoom Out


Navicat has the ability to zoom in or zoom out the SQL in the editor. The zooming options are available from the **View** menu. The same effect can be achieved with keyboard shortcuts.

- Zoom In: [Ctrl + =]
- Zoom Out: [Ctrl + -]
- Reset: [Ctrl + 0]

**Hint:** Range from -10 to +20.

**Note:** Files are opened in different tabs will not be effected by the zoom.

## Query Results

To run the query click  **Run** on the toolbar. If the query statement is correct, the query executes and, if the query statement is supposed to return data, the **Result** tab opens with the data returned by the query. If an error occurs while executing the query, execution stops, the appropriate error message is displayed.

The **Result** tab displays the result data, returned by the query, as a grid. Data can be displayed in three modes: **Grid View**, **Form View** and **Text/Blob View**. See Data View for details.

**Hint:** Navicat supports to return more than one resultset.

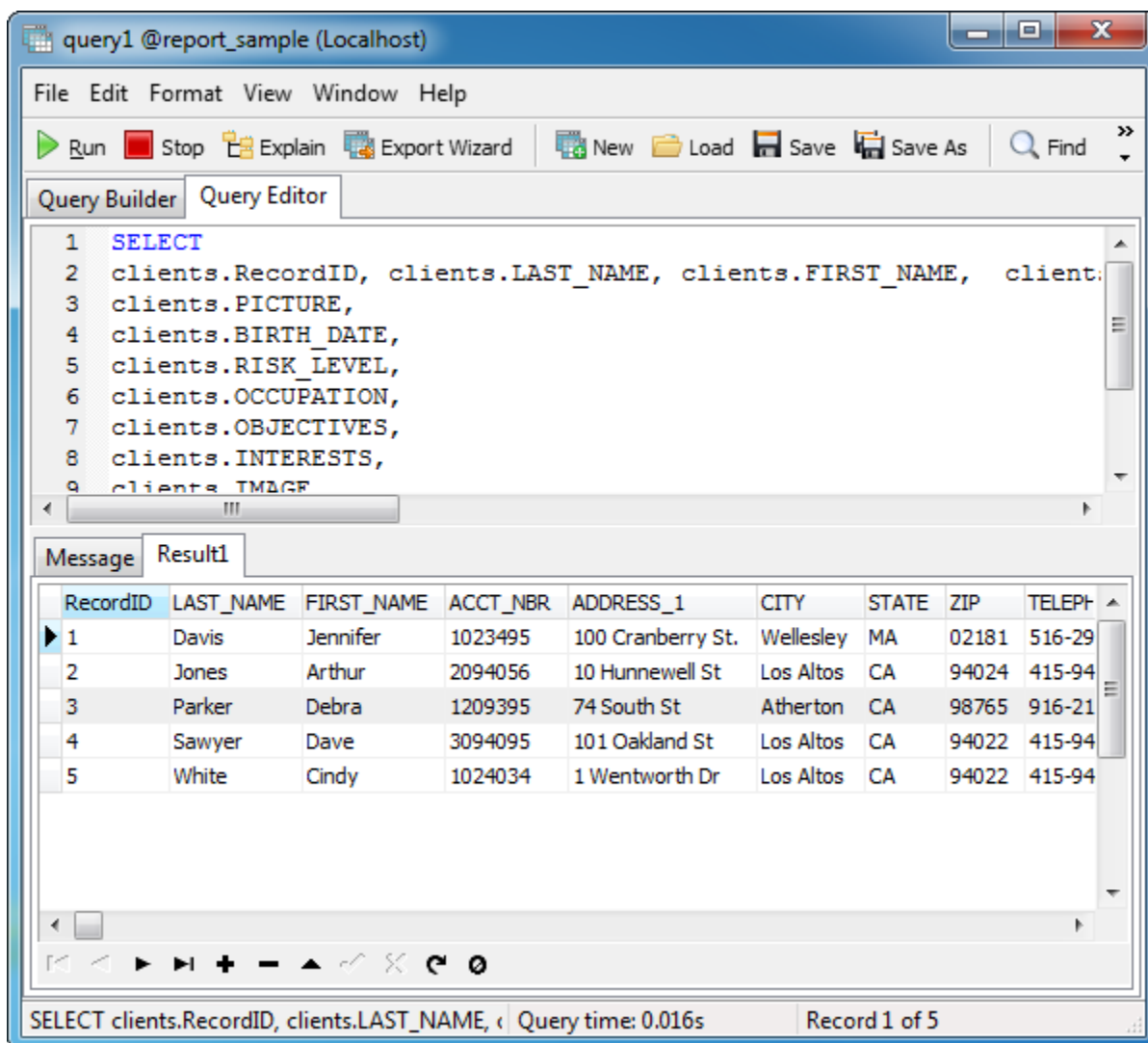
**Note:** You can choose to show the Result tab below the query editor or in a new tab by selecting View -> Show Result -> **Below Query Editor** or **In a New Tab**.

The toolbars of Query Result Viewer provides the following functions for managing data:


- Export Query Results  
Export data to MS Word, MS Excel, MS Access, TXT, DBF, HTML, SQL, RTF and more.
- Edit TEXT/BLOB/BFile  
Allow you to view and edit the content of TEXT/BLOB/BFile fields.

**Note:** Only Oracle supports BFile.





## Query Profile and Status (Available only for MySQL)

To show the profile and status when running the query, simply choose View -> **Show Profile and Status** and click  **Run** on the toolbar.

The **Profile** tab displays the query profile: Table lock, System lock, Statistic, etc.


**Note:** For MySQL 5.0, support from 5.0.37 or above.

For MySQL 5.1, support from 5.1.24 or above.

For MySQL 6.0, support from 6.0.5 or above.

The **Status** tab displays the query status: Bytes received, Bytes sent, etc.

## Query Explain

To show the Explain Plan of the query, just simply click  **Explain**.

- [Explain Plan for MySQL](#)
- [Explain Plan for Oracle](#)
- [Explain Plan for PostgreSQL](#)
- [Explain Plan for SQLite](#)
- [Explain Plan for SQL Server](#)

## Explain Plan for MySQL

The **Explain** tab displays the information as a grid:

Column	Description
id	The SELECT identifier. This is the sequential number of the SELECT within the query.
select_type	The type of SELECT, which can be SIMPLE, PRIMARY, UNION, DEPENDENT UNION, UNION RESULT, SUBQUERY, DEPENDENT SUBQUERY, DERIVED, UNCACHEABLE SUBQUERY or UNCACHEABLE UNION.
table	The table to which the row of output refers.
type	The join type.
possible_keys	The possible_keys column indicates which indexes MySQL can choose from use to find the rows in this table.
key	The key column indicates the key (index) that MySQL actually decided to use.
key_len	The key_len column indicates the length of the key that MySQL decided to use.
ref	The ref column shows which columns or constants are compared to the index named in the key column to select rows from the table.
rows	The rows column indicates the number of rows MySQL believes it must examine to execute the query.
filtered	The filtered column indicates an estimated percentage of table rows that will be filtered by the table condition.
Extra	This column contains additional information about how MySQL resolves the query.

## Explain Plan for Oracle

The **Explain** tab displays the data in the Oracle PLAN\_TABLE as a grid:

Column	Description
Operation	Name of the internal operation performed in this step.
Object	Name of the table or index.
Optimizer	Current mode of the optimizer.
Cost	Cost of the operation as estimated by the optimizer's query approach. Cost is not determined for table access operations. The value of this column does not have any particular unit of measurement; it is merely a weighted value used to compare costs of execution plans. The value of this column is a function of the CPU_COST and IO_COST columns.
Cardinality	Estimate by the query optimization approach of the number of rows accessed by the operation.
Bytes	Estimate by the query optimization approach of the number of bytes accessed by the operation.
Partition Start	Start partition of a range of accessed partitions.
Partition ID	Step that has computed the pair of values of the PARTITION_START and PARTITION_STOP columns.
Access Predicates	Predicates used to locate rows in an access structure. For example, start or stop predicates for an index range scan.
Filter Predicates	Predicates used to filter rows before producing them.

## Explain Plan for PostgreSQL

The **Message** tab displays the query plan.

## Explain Plan for SQLite

The **Explain** tab displays the query plan as a grid.

## Explain Plan for SQL Server

The **Explain** tab displays detailed information about how the statements are executed and provides estimates of the resource requirements for the statements.

Column	Description
StmtText	For rows that are not of type PLAN_ROW, this column contains the text of the Transact-SQL statement. For rows of type PLAN_ROW, this column contains a description of the operation. This column contains the physical operator and may optionally also contain the logical operator. This column may also be followed by a description that is determined by the physical operator.
StmtId	Number of the statement in the current batch.
NodeId	ID of the node in the current query.
Parent	Node ID of the parent step.
PhysicalOp	Physical implementation algorithm for the node. For rows of type PLAN_ROWS only.
LogicalOp	Relational algebraic operator this node represents. For rows of type PLAN_ROWS only.
Argument	Provides supplemental information about the operation being performed. The contents of this column depend on the physical operator.
DefinedValues	Contains a comma-separated list of values introduced by this operator. These values may be computed expressions which were present in the current query (for example, in the SELECT list or WHERE clause), or internal values introduced by the query processor in order to process this query. These defined values may then be referenced elsewhere within this query. For rows of type PLAN_ROWS only.
EstimateRows	Estimated number of rows of output produced by this operator. For rows of type PLAN_ROWS only.
EstimateIO	Estimated I/O cost* for this operator. For rows of type PLAN_ROWS only.
EstimateCPU	Estimated CPU cost* for this operator. For rows of type PLAN_ROWS only.



AvgRowSize	Estimated average row size (in bytes) of the row being passed through this operator.
TotalSubtreeCost	Estimated (cumulative) cost* of this operation and all child operations.
OutputList	Contains a comma-separated list of columns being projected by the current operation.
Warnings	Contains a comma-separated list of warning messages relating to the current operation. Warning messages may include the string "NO STATS:()" with a list of columns. This warning message means that the query optimizer attempted to make a decision based on the statistics for this column, but none were available. Consequently, the query optimizer had to make a guess, which may have resulted in the selection of an inefficient query plan.
Type	Node type. For the parent node of each query, this is the Transact-SQL statement type (for example, SELECT, INSERT, EXECUTE, and so on). For subnodes representing execution plans, the type is PLAN_ROW.
Parallel	0 = Operator is not running in parallel. 1 = Operator is running in parallel.
EstimateExecutions	Estimated number of times this operator will be executed while running the current query.


\* Cost units are based on an internal measurement of time, not wall-clock time. They are used for determining the relative cost of a plan in comparison to other plans.

## Query Parameters

Query Builder and Query Editor both support using of parameters inside the query text. You can set query parameters to add variable values to a query each time you run it. The parameter should appear as an identifier with **\$** at its beginning, quote with **[ ]**, e.g. `[$any_name]`.

Execute the query and the **Input Parameter** Dialog is provided for you to enter the desired data you wish to search.

## Debugging Oracle Query (Available only in Full Version)


To debug the Oracle query click  **Debug** on the toolbar to launch the Oracle Debugger.

Enter the parameter(s) if the query has input parameter(s).

## Data Modeling Tools (Available only in Navicat Premium and Enterprise Version)



**Model** is a powerful tool for creating and manipulating database models. Some of key features are listed here:

- Create and manipulate a model graphically.
- Reverse engineer a database/schema or table(s) to a model.
- Forward engineer a model to a sql file or database/schema.
- Create and edit table structures directly.

Just simply click  to open an object pane for **Model**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, edit and delete the selected model.

### Create Model

To create a new model

- Select anywhere on the object pane.
- Click the  **New Model** from the object pane toolbar.  
or
- Right-click and select  **New Model** from the popup menu.
- Edit model properties in the Model Designer.

To create a new model using reverse engineering




- Open a database/schema.
- Select the database/schema from the navigation pane or select the table(s) from the navigation pane/object pane.
- Right-click the database/schema/table(s) and select **Reverse Database to Model.../Reverse Tables to Model...** from the popup menu.
- Edit model properties in the Model Designer.

To create a new model with the same properties as one of the existing models has

- Select the model(s) for copying in the object pane.
- Right-click and select the **Copy** from the popup menu.
- Select anywhere on the object pane.
- Right-click and select the **Paste** from the popup menu.



- The newly created model(s) will be named as "modelname - **Copy**".

To create a new model with modification as one of the existing models

- Select the model for modifying in the object pane.
- Right-click and select the  **Design Model** from the popup menu.  
or
- Click the  **Design Model** from the object pane toolbar.
- Modify model properties in the Model Designer.
- Click  **Save As**.

## Edit Model

To edit the existing model



- Select the model for editing in the object pane.
- Right-click and select the  **Design Model** from the popup menu.  
or
- Click the  **Design Model** from the object pane toolbar.
- Edit model properties in the Model Designer.

To change the name of the model

- Select the model for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

## Delete Model

To delete a model

- Select the model for deleting in the object pane.
- Right-click and select the  **Delete Model** from the popup menu.  
or
- Click the  **Delete Model** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Model Information

To achieve a model information

- Select the model in the object pane.
- Right-click the selected model and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

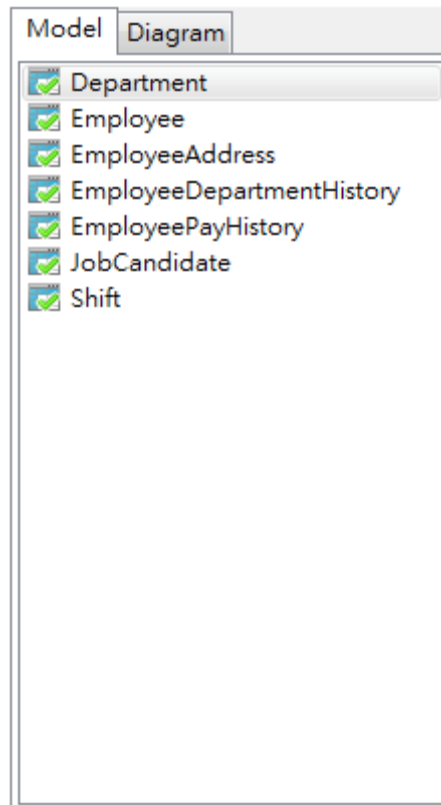
## Model Designer

**Model Designer** is the basic Navicat tool for working with model. It allows you to create, edit, delete, print model and etc.

- [Model Tree Palette](#)
- [Diagram Tree Palette](#)
- [Model Diagram Palette](#)
- [Properties Palette](#)
- [History Palette](#)
- [Model Preview and Print](#)

## Model Tree Palette

**Model Tree** holds all the tables in the model, including the tables used in each individual diagram. By default, it is located on the left sidebar. To show/hide the sidebar, simply choose View -> **Sidebar** from the main menu.



## Create Table

To create a new table

- Select anywhere in the model tree.
- Right-click and select **New Table** from the popup menu.

## Edit Table

To edit the existing table

- Right-click the table in the model tree and select **Design Table** from the popup menu.
- Edit table properties and fields on the appropriate tabs.



To change the name of the table

- Select the table for editing in the model tree.
- Right-click and select the **Rename** from the popup menu.

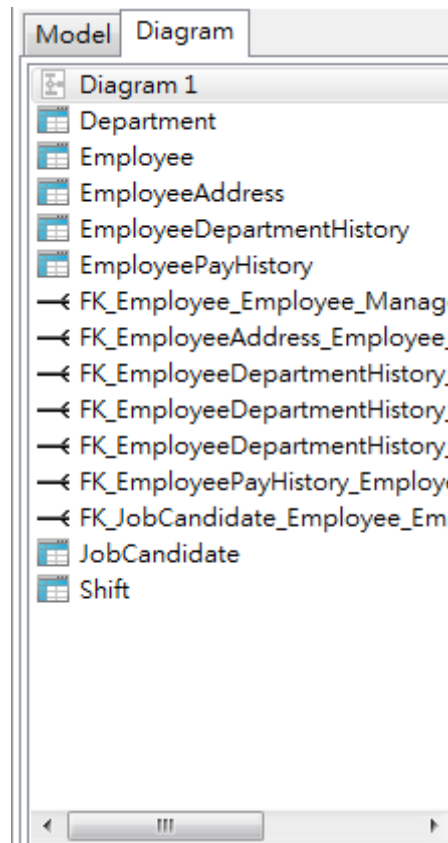
## Delete Table

To delete a table

- Select the table for deleting in the model tree.
- Right-click and select the **Delete Table** from the popup menu.
- Confirm deleting in the dialog window.

## Diagram Tree Palette

**Diagram Tree** holds all the objects (tables, layers, notes, images and relations) added to the active diagram. By default, it is located on the left sidebar. To show/hide the sidebar, simply choose View -> **Sidebar** from the main menu.



## Create Object

To create a new object

- Select anywhere in the diagram tree.
- Right-click and select **New** from the popup menu.
- Choose the object.

## Edit Object

To edit the existing table

- Right-click the table in the diagram tree and select **Design Table** from the popup menu.
- Edit table properties and fields on the appropriate tabs.

To change the name of the object

- Select the object for editing in the diagram tree.
- Right-click and select the **Rename** from the popup menu.

## Delete Object

To delete a table from the diagram

- Select the table for deleting in the diagram tree.
- Right-click and select the **Delete -> from Diagram** from the popup menu.
- Confirm deleting in the dialog window.


To delete a table/relation from the model and diagram

- Select the table/relation for deleting in the diagram tree.
- Right-click and select the **Delete -> from Diagram and Model** from the popup menu.
- Confirm deleting in the dialog window.

To delete a layer/note/image from the diagram


- Select the object for deleting in the diagram tree.
- Right-click and select the **Delete** from the popup menu.
- Confirm deleting in the dialog window.

## Model Diagram Palette

**Model Diagram Palette** is a canvas for you to design the diagram. Simply click the  **New Diagram** from the toolbar to create a new diagram.

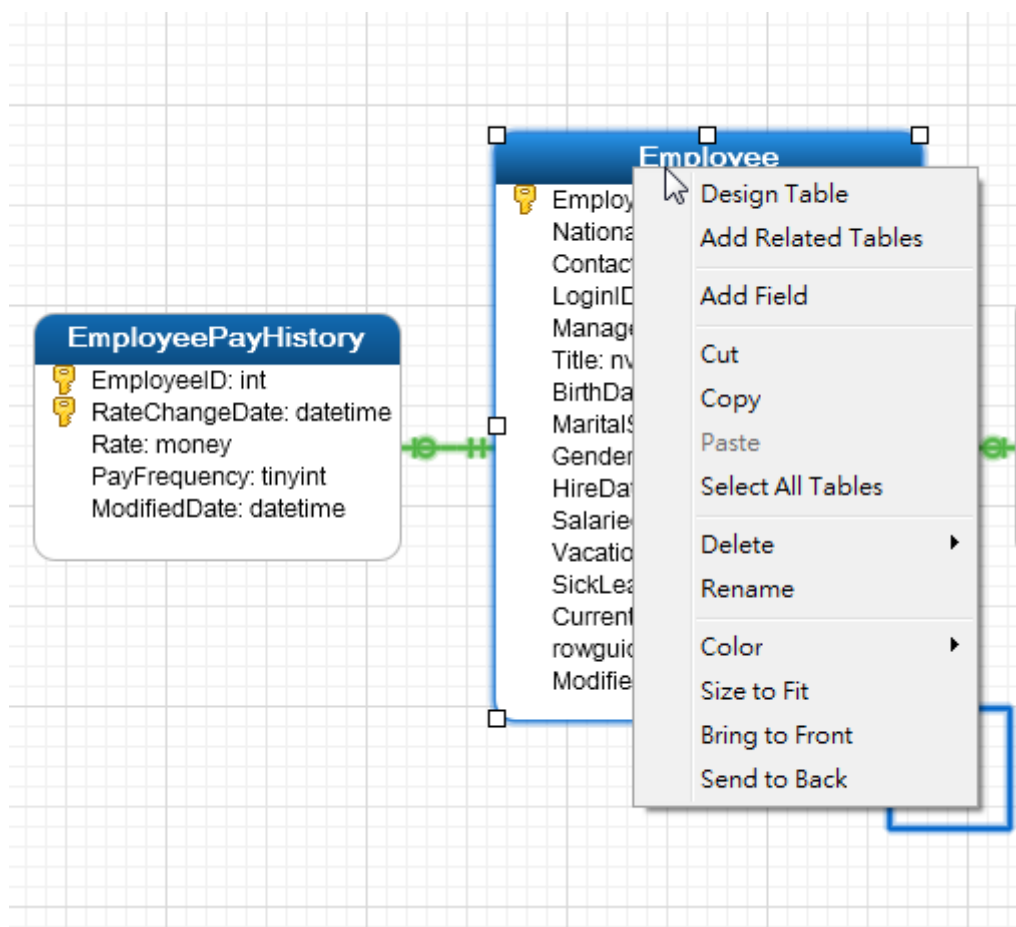
- [Creating Tables](#)
- [Creating Layers](#)
- [Creating Notes and Labels](#)
- [Creating Images](#)
- [Creating Relations](#)
- [Formatting Diagram](#)

## Creating Tables

To create a new table, click the  button from the diagram toolbar and click anywhere on the canvas.

To add an existing table from the model tree to the diagram, simply drag and drop the table from model tree to the canvas.

**Note:** If you right-click a field in a table, you can choose to add, insert, delete, rename field and set the field as primary key.



The popup menu options of the table object in canvas include:

### Design Table

Edit the table structure in a designer, e.g. fields, indexes, foreign keys, etc.

**Note:** The tabs and options in the designer depend on the diagram database type you are chosen. For the settings of different tabs, see Database Object Management.

## **Add Related Tables**

Add all related tables to the selected table.

## **Add Field**

Add fields to the existing table.

## **Cut**

Remove the table from the diagram and put it on the clipboard.

## **Copy**

Copy the table from the diagram to the clipboard.

## **Paste**

Paste the content from the clipboard into the diagram.

## **Select All Tables**

Select all the tables in the diagram.

## **Delete**

Delete a table from the diagram or from both diagram and model.

## **Rename**

Change the name of the table.

## **Color**

Change the color of the table.

## **Size to Fit**

Resize the table automatically to fit its contents.

## **Bring to Front**


Bring table to the foreground.

## **Send to Back**

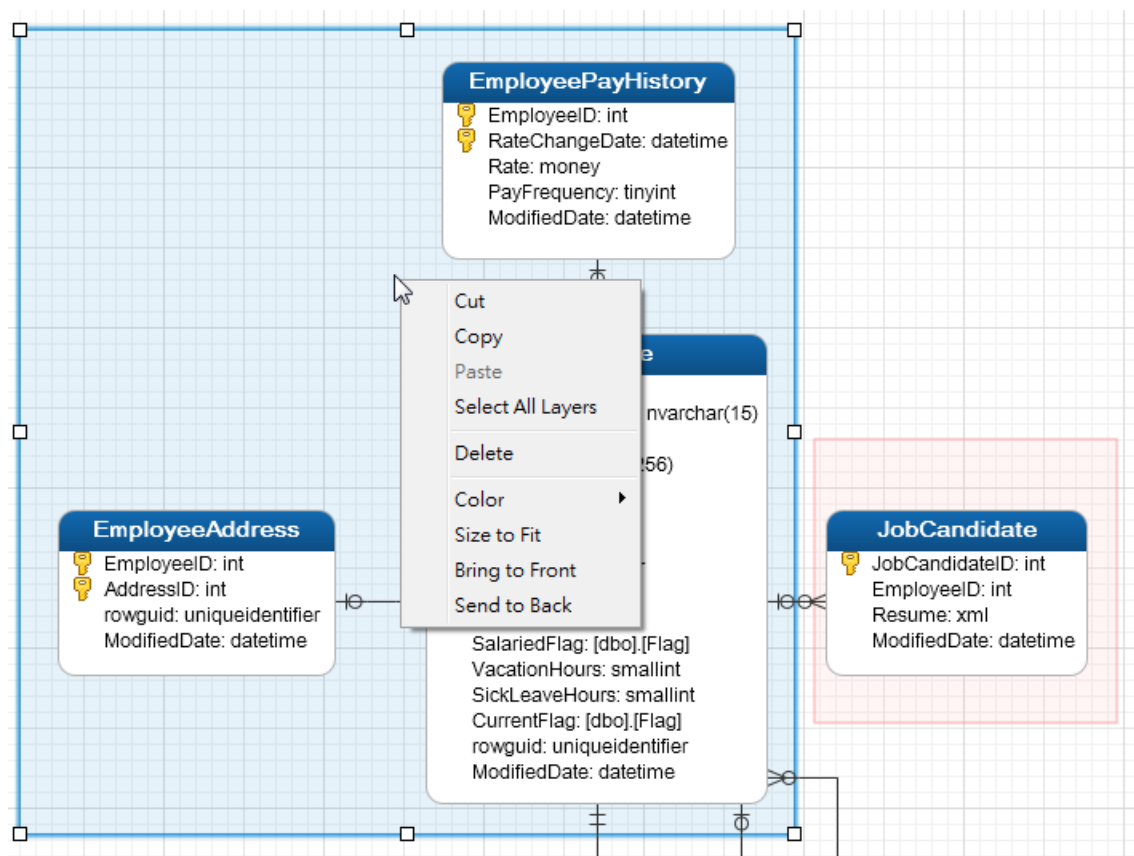
Move table to the background.

## Creating Layers

Layers are used to help organize objects (e.g. tables, notes, images, etc) on the canvas. You can add all related objects to the same layer. For example, you may choose to add all your sales related tables to one layer.

To create a new layer, click the  button from the diagram toolbar and click anywhere on the canvas.

You can put any objects on top of the layer.



The popup menu options of the layer object in canvas include:

### Cut

Remove the layer from the diagram and put it on the clipboard.

### Copy

Copy the layer from the diagram to the clipboard.

**Paste**

Paste the content from the clipboard into the diagram.

**Select All Layers**

Select all the layers in the diagram.

**Delete**

Delete a layer from the diagram.

**Color**

Change the color of the layer.

**Size to Fit**

Resize the layer automatically to fit its contents.

**Bring to Front**

Bring layer to the foreground.


**Send to Back**

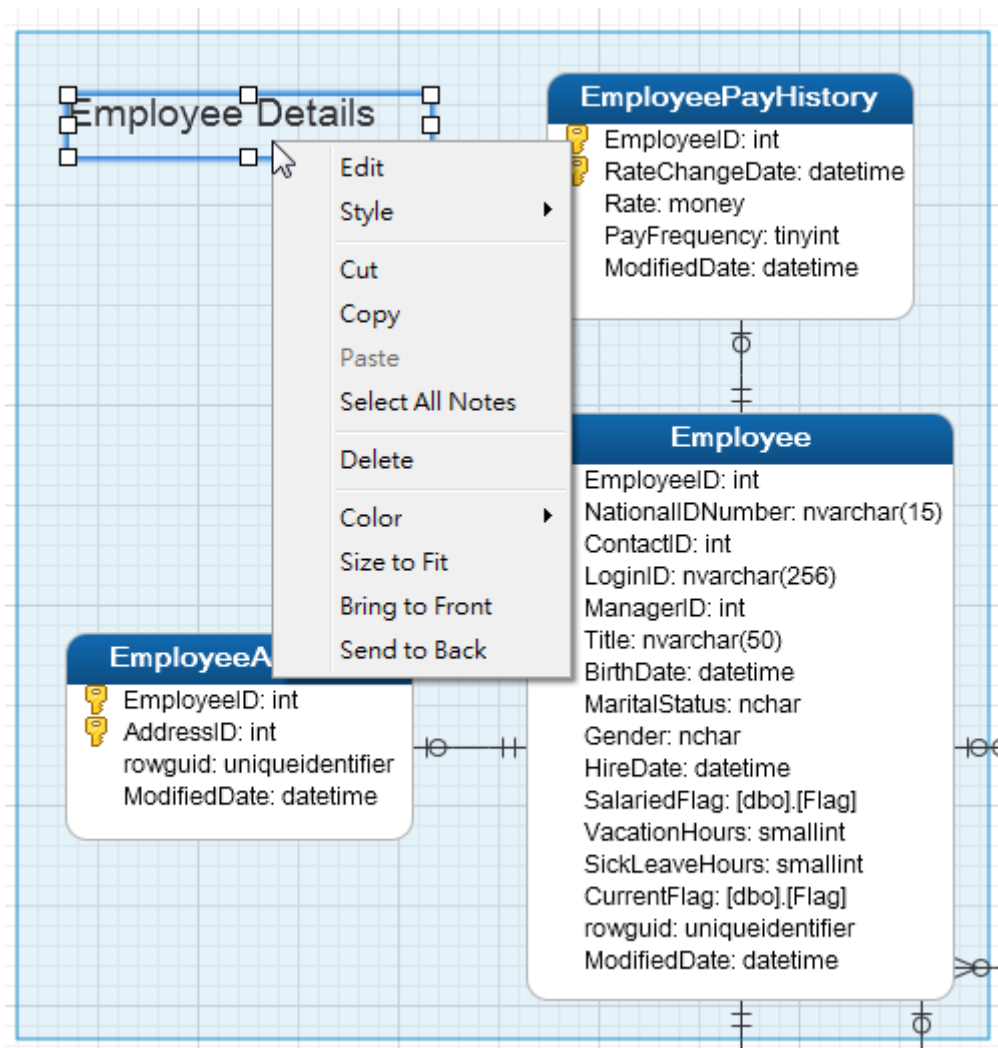
Move layer to the background.



## Creating Notes and Labels

Notes or Labels are typically used to help document the diagram design process. For example, to explain a grouping table objects.

To create a new note or label, click the  button from the diagram toolbar and click anywhere on the canvas.



The popup menu options of the note object in canvas include:

### Edit

Change the content of the note.

### Style

Choose the style of the note: Note or Label.

**Cut**

Remove the note from the diagram and put it on the clipboard.

**Copy**

Copy the note from the diagram to the clipboard.

**Paste**

Paste the content from the clipboard into the diagram.

**Select All Notes**

Select all the notes in the diagram.

**Delete**

Delete a note from the diagram.

**Color**

Change the color of the note.

**Size to Fit**

Resize the note automatically to fit its contents.


**Bring to Front**

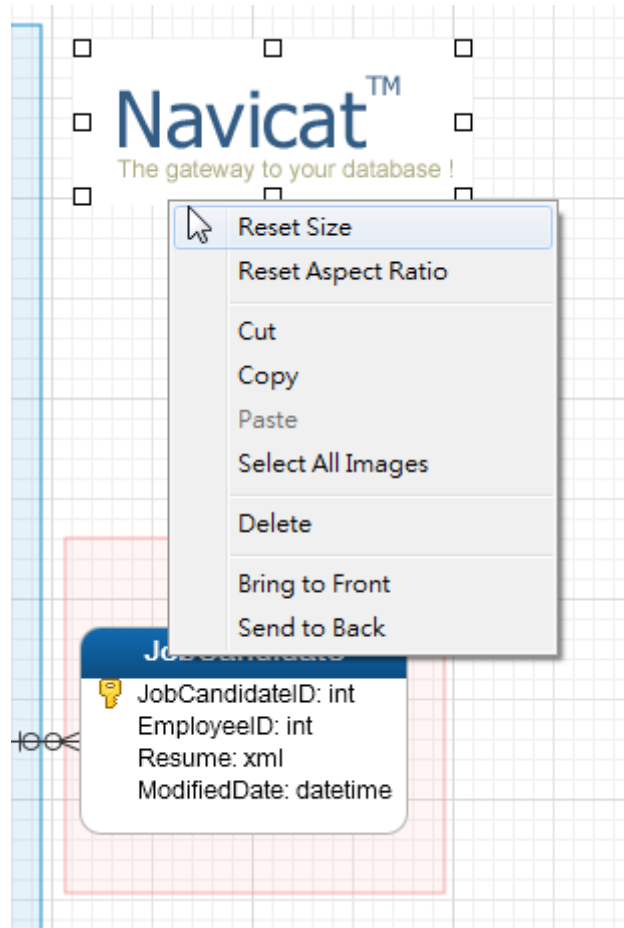
Bring note to the foreground.

**Send to Back**

Move note to the background.

## Creating Images

To create a new image, click the  button from the diagram toolbar and click anywhere on the canvas.



The popup menu options of the image object in canvas include:

### **Reset Size**

Reset the size of the image to its original size.

### **Reset Aspect Ratio**

Maintain image original width to height ratio.

### **Cut**

Remove the image from the diagram and put it on the clipboard.

### **Copy**

Copy the image from the diagram to the clipboard.

**Paste**

Paste the content from the clipboard into the diagram.

**Select All Images**

Select all the images in the diagram.

**Delete**

Delete a image from the diagram.


**Bring to Front**

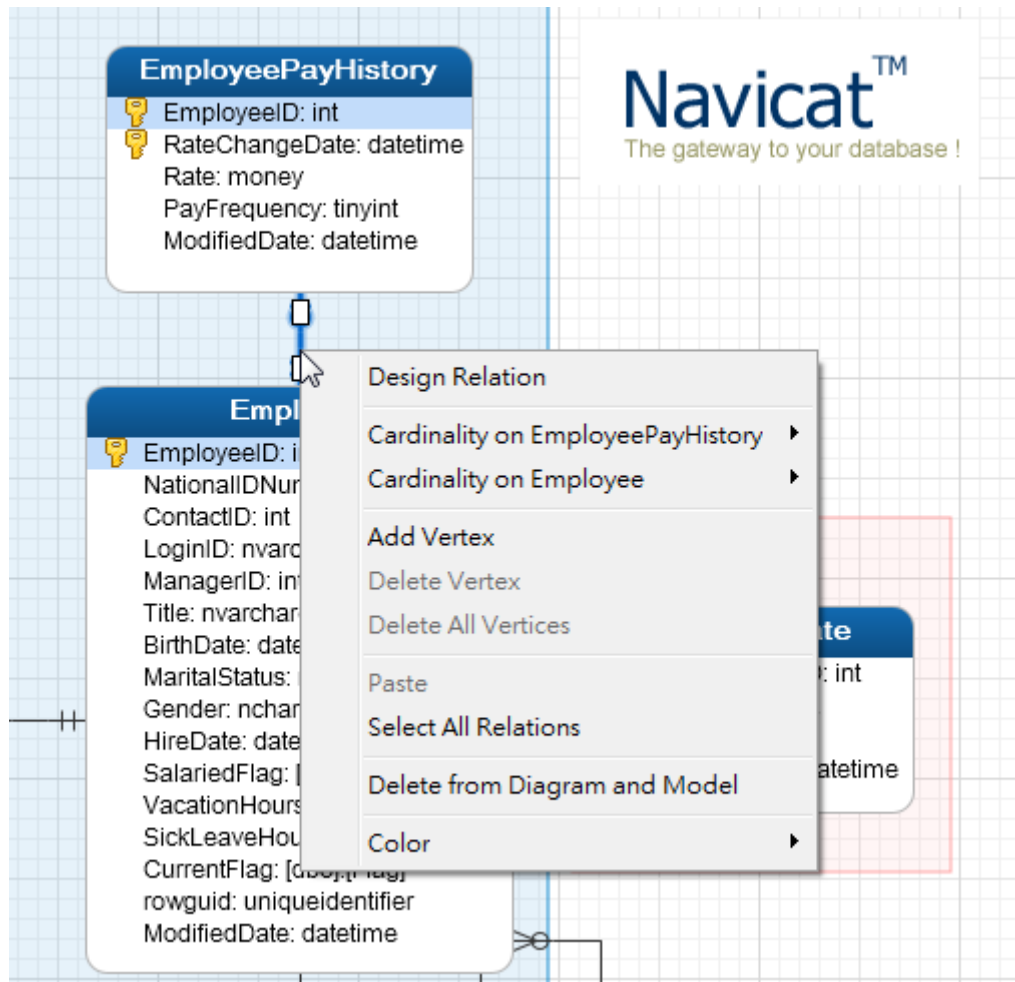
Bring image to the foreground.

**Send to Back**

Move image to the background.

## Creating Relations

To add a relation, click the  button from the diagram toolbar and drag and drop a field from one table to another.



The popup menu options of the relation object in canvas include:

### Design Relation

Edit the relation in a designer.

**Note:** The options in the designer depends on the diagram database type you are chosen. For the settings, see Database Object Management.

### Cardinality on table\_name1

Set the cardinality on table\_name1: None, One and Only One, Many, One or Many, Zero or One, Zero or Many.

## **Cardinality on table\_name2**

Set the cardinality on table\_name2: None, One and Only One, Many, One or Many, Zero or One, Zero or Many.

## **Add Vertex**

Add a vertex on a relation connector.

## **Delete Vertex**

Delete a vertex on a relation connector.

## **Delete All Vertices**

Delete all vertices on a relation connector.

## **Paste**

Paste the content from the clipboard into the diagram.

## **Select All Relations**

Select all the relations in the diagram.

## **Delete from Diagram and Model**

Delete a relation from the diagram and model.

## **Color**

Change the color of the relation.

## Formatting Diagram

### Show Grid

To turn the grid on in the diagram canvas, choose View -> **Show Grid** from the main menu.

### Snap To Grid

To align objects on the canvas with the grid, choose View -> **Snap To Grid** from the main menu.

### Change Database Type

To change the database type of the diagram, choose Model -> **Database** and select the type from the main menu. Choose **Generic** if you do not decide the database type.

### Change Diagram Notation

To change the notation of the diagram, choose Model -> **Diagram Notation** and select the notation from the main menu.

#### Default

The default notation style used in Navicat.

#### Simple

A simple notation style. The table objects will only show the name.

#### Crow's Foot

Crow's Foot notation style.

#### IDEF1X

The ICAM DEFinition language information modeling method.

#### UML

Universal Modeling Language style.


#### Black and White

Change the color of the diagram to black and white.

### Change Diagram Dimensions

To change the number of pages used in the diagram, choose Model -> **Diagram Dimensions...** from the main menu and set the Width and Height.

## Apply Auto Layout

To automatically arrange objects on the canvas, click  **Auto Layout** from the toolbar. To change the Auto Layout, simply choose Tools -> **Auto Layout with...** from the main menu and set the options.

### Auto Diagram Dimension

Choose the suitable diagram dimension automatically.

### Auto Size Tables to Fit

Resize the table to fit its content automatically.

### Quality

The quality of the auto layout output.

### Object Distance

The distance between the objects in the diagram.

## Zoom In/Zoom Out

To zoom in or zoom out the selected area of the diagram, adjust the slider of the **Navigator**. If the Navigator is hidden, choose View -> **Navigator** from the main menu. Same effect can be achieved with keyboard shortcuts:

Zoom In: [Ctrl++ ] or [Ctrl+Mousewheel up]

Zoom out: [Ctrl+- ] or [Ctrl+Mousewheel down]



## Properties Palette

The **Properties** palette is used to display and edit the properties of diagram and its objects.

### Black and White

Check this box to change the diagram color to black and white.

### Bold

Check this box or press Ctrl+B to bold the table or relation.

### Color

The color of the object.

### Diagram Notation

The notation of the diagram. The value for this can be Default, Simple, IDEF1X, UML or Crow's Foot.

### Font Bold

Check this box to bold the note font.

### Font Color

The font color of the note.

### Font Italic

Check this box to apply an italic style to the note font.

### Font Name

The font name of the note.

### Font Size

The font size of the note.

### Height

The height of the object.

### Height (pages)

The height of the diagram (number of papers).

### Left

The number of pixels from the object to the left side of the canvas.

**Name**

The name of the object.

**Opacity**

The transparency of the image. The value for this can be between 0 to 100. Use 100 for opacity and 0 for transparent.

**Referenced Cardinality**

The relation cardinality of referenced (parent) table.

**Referencing Cardinality**

The relation cardinality of referencing (child) table.

**Style**

The style of the note. The value for this can be Note or Label.

**Table Font Name**

The font name of the tables.

**Table Font Size**

The font size of the tables.

**Top**

The number of pixels from the object to the top of the canvas.

**Width**

The width of the object.

**Width (pages)**


The width of the diagram (number of papers).

## History Palette

The **History** palette shows all the actions that you have taken. Simply double-click a action to restore that state.

## Model Preview and Print


### Print Preview

To preview the pages before printing, simply click the  **Print Preview** button. The model can be printed to the printer or to various file formats.

### Page Setup

Choose File -> **Page Setup** to change paper size, orientation and margins.

### Print to a printer

Choose File ->  **Print** to send your diagram directly to the printer. You can set the printer option in the popup window.

### Print to a file (PDF/PNG/SVG)

Choose File -> **Print As** -> **PDF/PNG/SVG** to create a PDF/PNG/SVG file of your diagram.

## Export SQL

**Export SQL** allows exporting table structures in model into a SQL file.

- Select **Tools** -> **Export SQL...** from the main menu.
- Edit export properties on the appropriate tabs.
- Click **OK**.

## General Settings for Export SQL

### File

Set the output file name and location.

### Tables

You are allowed to choose your preferable tables in model you wish to export.

## Advanced Settings for Export SQL

### Server Version

Select server version for the SQL file.

#### ☒ **Include Schema (Available only for Oracle, PostgreSQL and SQL Server)**

Includes entered schema name in file with this option is on. Otherwise, only table name is included in SQL statements.

#### ☒ **Include Drop SQL**

Includes drop table SQL statements in file with this option in on.

#### ☒ **Include Drop with CASCADE (Available only for MySQL, Oracle and PostgreSQL)**

Includes drop table SQL statements with cascade option in file with this option in on.

#### ☒ **Include Primary Keys**

Includes primary keys in file with this option is on.

#### ☒ **Include Foreign Keys**

Includes foreign keys in file with this option is on.

#### ☒ **Include Uniques (Available only for Oracle, PostgreSQL, SQLite and SQL Server)**

Includes uniques in file with this option is on.

#### ☒ **Include indexes**

Includes indexes in file with this option is on.

#### ☒ **Include character set (Available only for MySQL)**

Includes table and field character set in file with this option is on.

#### ☒ **Include Collation (Available only for SQLite and SQL Server)**

Includes table collation in file with this option is on.

## Synchronize to Database

Navicat allows you to compare and modify the table structures with detailed analytical process. In other words, Navicat compares tables between the target databases/schemas and the model and states the differential in structure.

- Select **Tools** -> **Synchronize to Database...** from the main menu.
- Edit synchronization properties on the General tab.
- Click **Compare** to generate a set of scripts which shows the differentiation between the databases/schemas and the model.
- Select the scripts you want to run.
- Click **Run Query**.



## General Settings for Synchronize to Database

The following instruction guides you through the process of setting up a synchronization. Customize options according to your needs.

### Target

Defines connection, database and schema for the target.

### Compare Options

☒ **Compare Identifier With Case Sensitive**

Check this option if you want to compare table identifier with case sensitive option.

☒ **Compare Primary Keys**

Check this option if you want to compare table primary keys.

☒ **Compare Foreign Keys**

Check this option if you want to compare table foreign keys.

☒ **Compare Indexes**

Check this option if you want to compare indexes.

☒ **Compare Character Set (Available only for MySQL)**

Check this option if you want to compare character set of the tables.

☒ **Compare Uniques (Available only for Oracle, PostgreSQL, SQLite and SQL Server)**

Check this option if you want to compare uniques.

☒ **Compare Collation (Available only for SQLite and SQL Server)**

Check this option if you want to compare collation of the tables.

### Execution Options

☒ **SQL for objects to be created**

Check this option to include all related SQL statements if new objects will be created in the target database.

☒ **SQL for objects to be changed**

Check this option to include all related SQL statements if objects will be changed in the target database.

☒ **SQL for objects to be dropped**

Check this option to include all related SQL statements if objects will be dropped from the target database.

☒ **Continue on error**

Ignores errors that are encountered during the synchronization process.

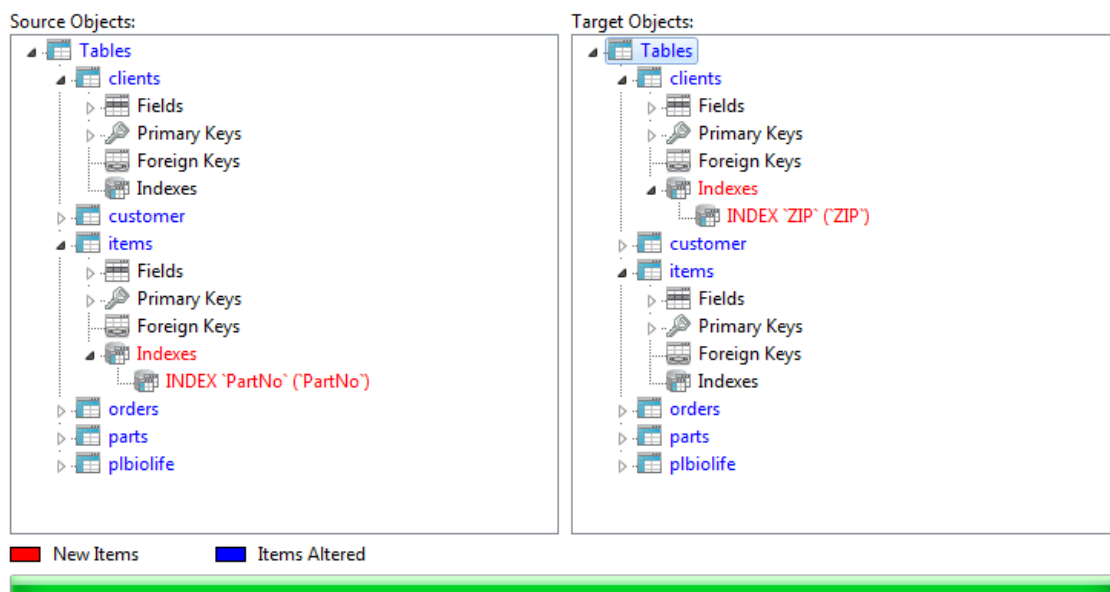
## Synchronize to Database Result

### Source Objects/Target Objects

The tree view shows the differentiation between the source model and target database/schema after the computation of the structure synchronization, providing with the detailed SQL statements shown in the **Queries for Modification** list.

The red item represents the non-existence for the other database/schema.

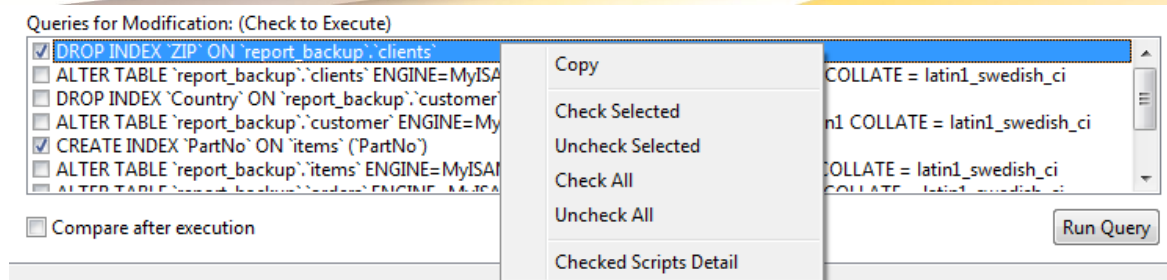
The blue item represents the existence for the other database/schema, but different definition detected.



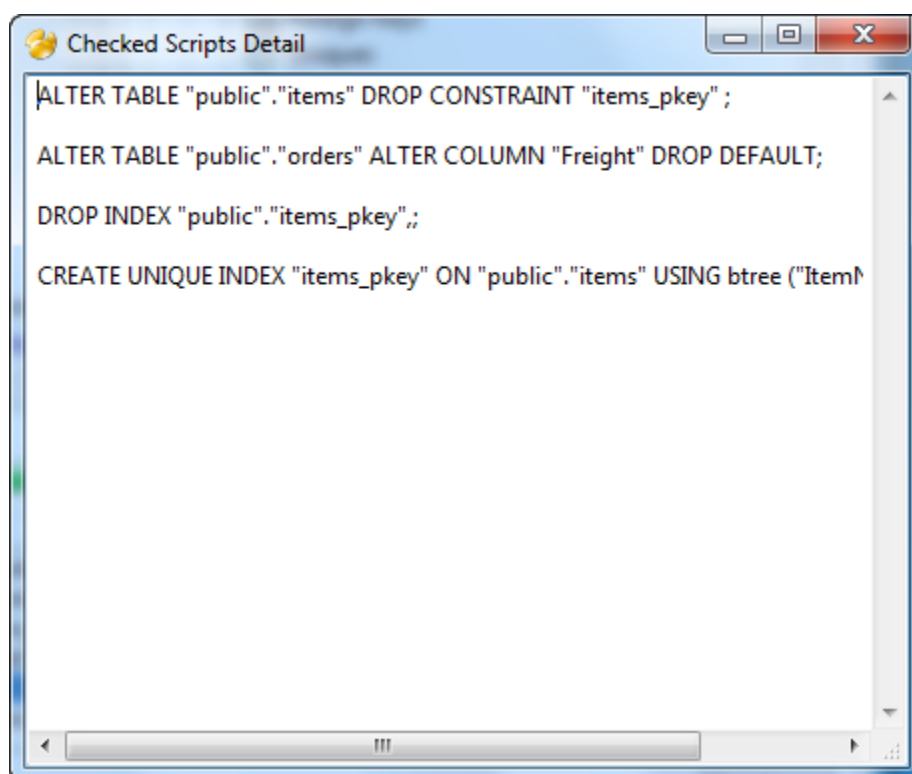
### Queries for Modification

All the scripts are applied to the target database/schema and they are being unchecked in the **Queries for Modification** list by default. Just simply select the scripts you want to execute.

You can highlight multiple lines of scripts, and then right-click to show the pop-up menu. Choose **Copy** can copy the selected queries to preferred editor. Choose **Check Selected**, **Uncheck Selected**, **Check All** or **Uncheck All** so as to perform selection/unselection of scripts at one go.



To view the full SQL statements you selected, right-click the **Queries for Modification** list and select **Checked Scripts Detail**.



## ☒ **Compare after execution**

Compares tables after the synchronization is executed.

Press **Run Query** to execute the selected query.

## Synchronize to Database Message Log

The **Message Log** tab allows you to view the running process indicating success or failure.



Example:

```
[Msg] Starting Synchronization
[Msg] Target Server : MySQL 5.1
[Msg] Target Database : report_sample
[Msg] Executing - ALTER TABLE `report_sample`.`orders` ADD CONSTRAINT
`fk_orders_customer_1` FOREIGN KEY (`CustNo`) REFERENCES `customer` (`CustNo`)
[Msg] Completed
[Msg] Synchronization Completed
```

## Model Hints and Tips

Navicat provides some useful hints to work on the model more effectively.

Action	Description
Locate Object in Model Diagram Palette	<ul style="list-style-type: none"> <li>- Object selected in Diagram Tree will be highlighted in Model Diagram.</li> <li>- Double-click an object in Diagram Tree will jump to the corresponding object.</li> </ul>
Delete Table from Model Tree Palette	<ul style="list-style-type: none"> <li>- Select table in Model Diagram and press Shift + Delete.</li> </ul>
Open Table Designer	<ul style="list-style-type: none"> <li>- Double-click a table in Model Tree/Model Diagram.</li> </ul>
Add table from Navicat Main	<ul style="list-style-type: none"> <li>- Drag table from Navicat main window and drop to the Model Diagram.</li> </ul>
Get Table Structure (SQL Statement)	<ul style="list-style-type: none"> <li>- Select and copy the table in Model Diagram, and paste it to text editor, e.g. Notepad.</li> </ul>
Design Field without Table Designer	<ul style="list-style-type: none"> <li>- Select and click the table name and press Tab/Down Arrow to add/edit fields.</li> </ul> <p>Navicat will predict field types according to field names you entered.</p> <p><b>INTEGER/int/int4/NUMBER</b></p> <ul style="list-style-type: none"> <li>- suffix "id", "no" (if it is the first column, it will be predicted as a primary key)</li> <li>- suffix "num"</li> <li>- "qty", "number"</li> <li>- exactly "age", "count"</li> </ul> <p><b>DECIMAL(10,2)/decimal(10,2)/NUMBER/REAL/money</b></p> <ul style="list-style-type: none"> <li>- suffix "price", "cost", "salary"</li> </ul> <p><b>FLOAT/double/float8/NUMBER/REAL/float</b></p> <ul style="list-style-type: none"> <li>- "size", "height", "width", "length", "weight", "speed", "distance"</li> </ul> <p><b>DATE/datetime/date/TEXT/datetime2</b></p> <ul style="list-style-type: none"> <li>- "date", "time"</li> </ul>

	<p><b>VARCHAR(255)/varchar(255)/VARCHAR2(255)/TEXT</b></p> <p>- other field names</p> <p>Enter * before the field name to recognize as primary key. e.g. *itemNo:int.</p> <p>Enter : between field name and field type to custom field type, e.g. itemName:varchar(255).</p>
Reorder Field	<p>- Select table in Model Diagram, then press and hold the Shift key.</p> <p>Use  to drag the field to a desired location.</p>
Delete Field	<p>- Select table in Model Diagram, then press and hold the Shift key.</p> <p>Use  to drag the desired field out of the table.</p>
Add Vertex to Relation	<p>- Select relation in Model Diagram, then press and hold the Shift key. Click on the relation to add vertex.</p>
Delete Vertex on Relation	<p>- Select relation in Model Diagram, then press and hold the Shift key. Click on the vertex.</p>
Delete Relation from Model	<p>- Select relation in Model Diagram and press Shift + Delete.</p>
Switch to Hand Mode	<p>- Press and hold the Space key, then move the diagram.</p>
Select a Page in Print Preview	<p>- Press and hold the Shift key, then point to a page to show the page number.</p> <p>- Press and hold the Shift key, then click a page to jump to the corresponding page in Model Diagram.</p>

## Data Management Tools

Navicat provides a number of powerful tools for working with the databases.

The following tools are available:

### [Import Wizard](#)

Imports data from DBF, TXT, CSV, HTML, Excel, Access, XML, ODBC and more.

### [Export Wizard](#)

Exports data to various formats, including DBF, TXT, CSV, HTML, Word, Excel, Access, XML, RTF and more.

### [Data Transfer](#)

Transfers tables/views/procedures/functions/sequences/events between databases/schemas or to plain text file.

### [Data Synchronization](#)

Synchronizes data in different databases/schemas to be kept up-to-date so that each repository contains the same information.

### [Structure Synchronization](#)

Compares the structure of two similar databases/schemas and produces a set of alter statements for MySQL, Oracle, PostgreSQL and SQL Server.

### [Backup/Restore](#)

Allows you to backup/restore your databases/schemas for MySQL, PostgreSQL and SQLite.

### [Batch Job/Schedule](#)

Allows you to schedule a batch job which being executed at a specified time and support e-mail notification service.

### [Console](#)

Provides interactive text-based screen for user query input and result output from MySQL, Oracle, PostgreSQL, SQLite and SQL Server.

### [Dump SQL File](#)

Dumps database/schema/table(s) to SQL file.



## [Execute SQL File](#)

Executes SQL file.

## [Print Structure](#)

Prints database/schema/table structure.

## [Log Files](#)

Keeps track on the actions (e.g. SQL statements being executed) which have been performed in Navicat.

## Import Wizard

**Import Wizard** allows you to import data to a table from DBF, TXT, CSV, HTML, Excel, Access, XML, ODBC and more. You can save your settings as a profile for setting schedule.

**Note:** Navicat Essentials version supports to import text-based files, such as TXT, CSV, HTML and XML file.

**Note:** You can drag a supported file to the table pane or a database/schema in the connection tree. Navicat will popup the import wizard. (If existing table is highlighted, Navicat will import the file to the highlighted table, otherwise, import the file to a new table)

To open the Import Wizard, click  **Import Wizard** from the table object pane toolbar.

- [Setting Import File Format \(Step 1\)](#)
- [Selecting Source File Name \(Step 2\)](#)
- [Setting Additional Options for Specific File Type](#)
- [Setting Target Table \(Step 5\)](#)
- [Adjusting Field Structures and Mapping Fields \(Step 6\)](#)
- [Selecting Import Mode \(Step 7\)](#)
- [Saving and Confirming Import \(Step 8\)](#)

To run a saved import profile from the command line

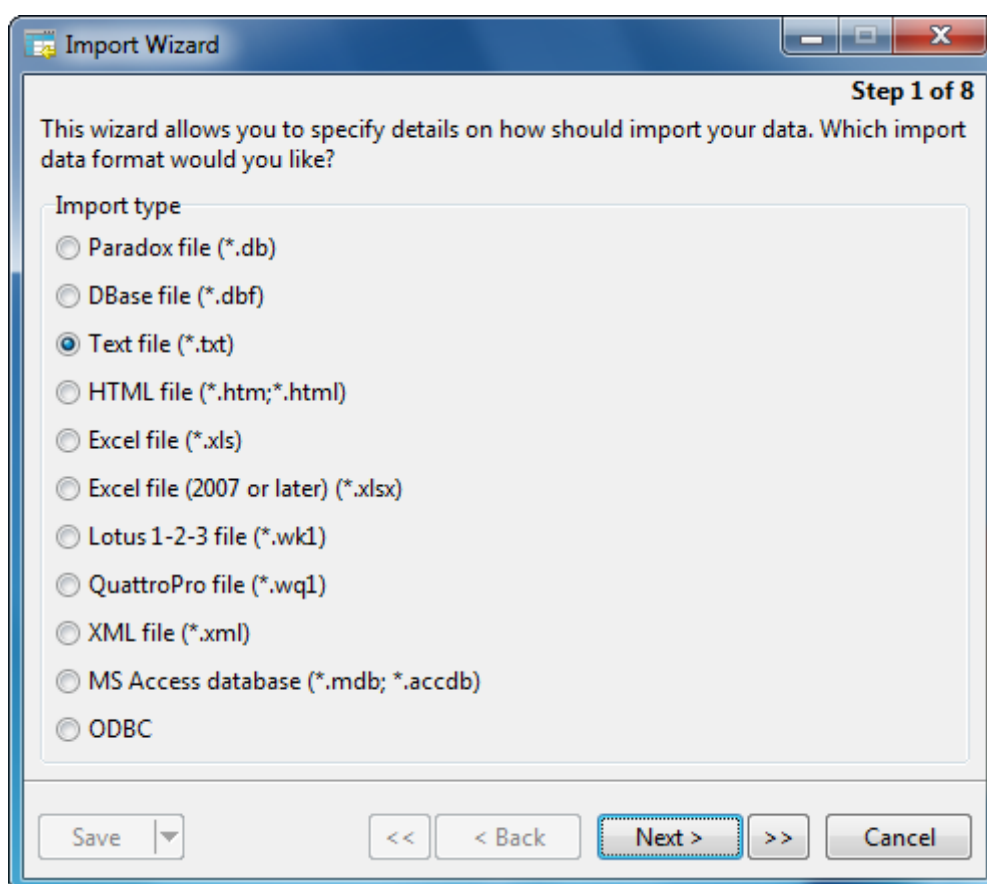
- Create and save the import profile.
- Start Navicat from command line, type the command (see Command for details)

## Setting Import File Format (Step 1)

Select one of the available import types for the source file.

**Note:** Navicat Essentials only supports importing from TXT, CSV, HTML and XML file.

**Note:** The Excel file format is according to the Microsoft Office version installed in your computer.



## Setting Source File Name (Step 2)

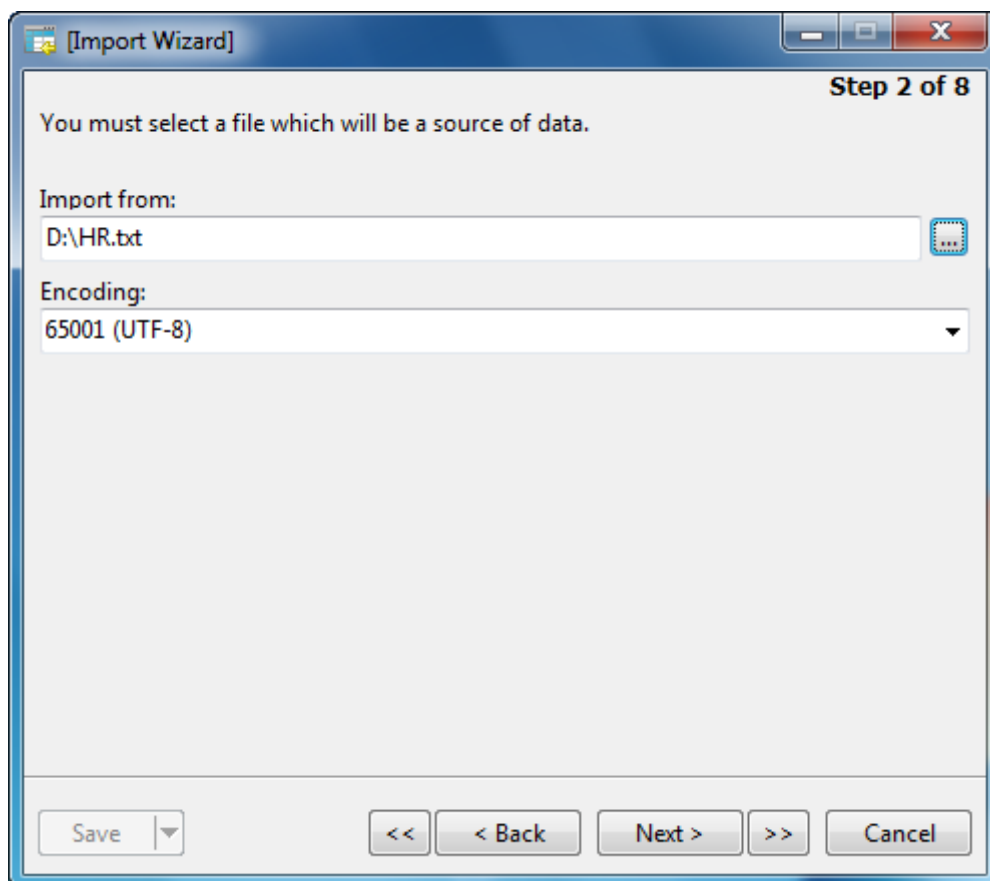
### Import from

Browse the source file name. The file name extension in the Import from text box changes according to the selected table type in step 1.

**Note:** For TXT and XML file, you can select more than one file to import.

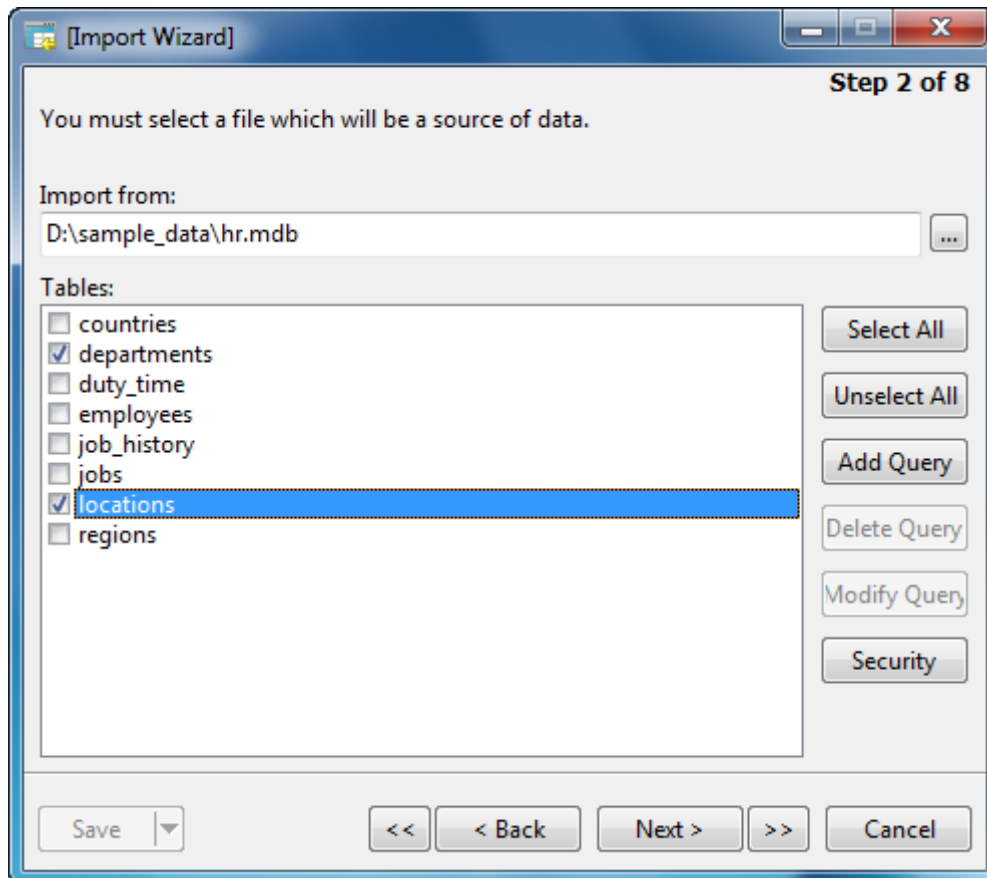
### Encoding

Select the encoding for the source file.



For importing Excel and Access, a list of sheets (Excel), tables and queries (Access) will be shown in the list below.

**Hint:** **Add Query**, **Delete Query** and **Modify Query** are only available on Access/ODBC import styles.



## Security

If there is security settings, i.e. database password and user level security in your access file, you are required to input the necessary information.

### System Database File

Locate the system security file of the Access file, e.g. D:\Temp\Security.mdw.

### Database Password

Enter the password for the database if any.

### Logon Name

Enter the user name set by the user level security.

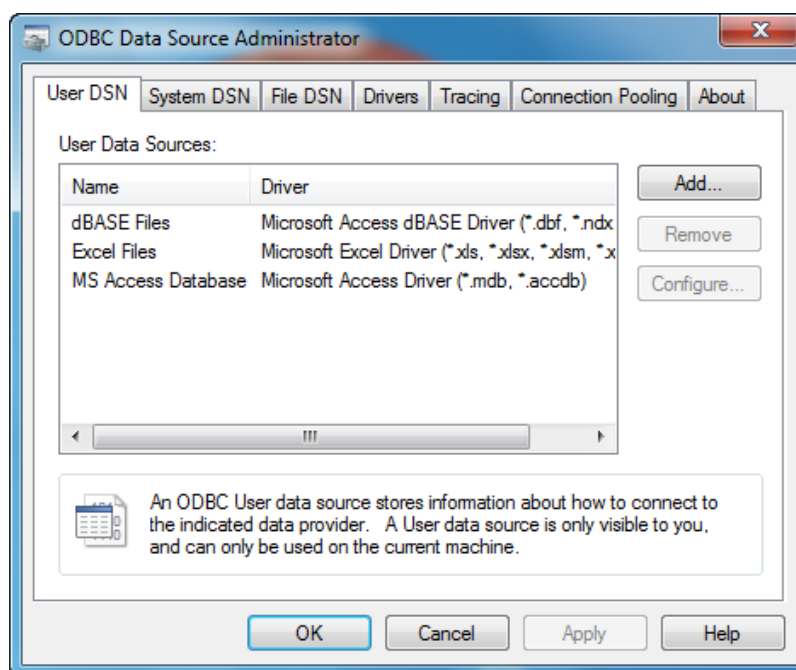
### Logon Password

Enter the password of that user.

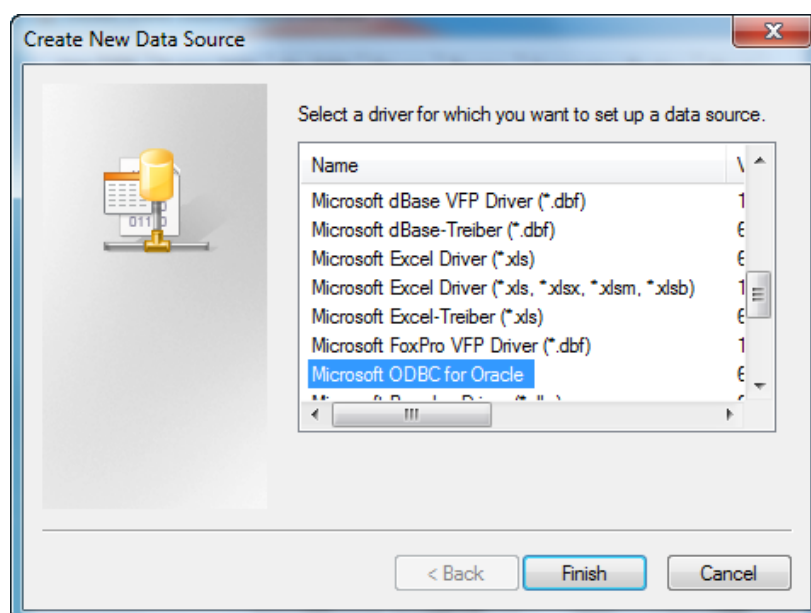
## Importing ODBC Data (Step 2)

### Setting Up an ODBC Data Source Connection

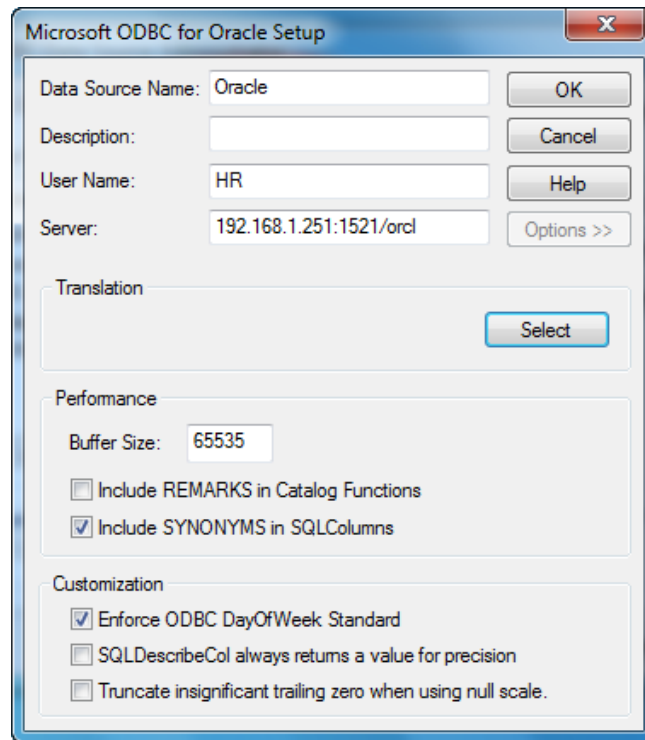
1. On the Control Panel, select **Administrative Tools**.
2. Select **Data Sources (ODBC)**.
3. Select **User DSN** tab.



4. Click **Add**.
5. Select the correct ODBC driver you wish, such as Oracle and click **Finish**.

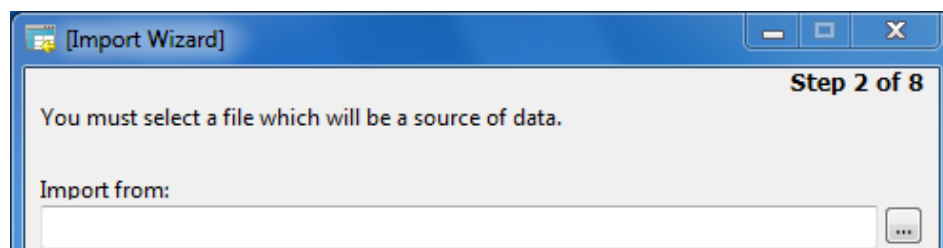


6. Type a meaningful name for this ODBC data source in the **Data Source Name** text box.
7. Type a description for the data source in the **Description** text box.
8. Type server name in the **Server** text box.
9. Select **OK** to see your ODBC Driver in the list.

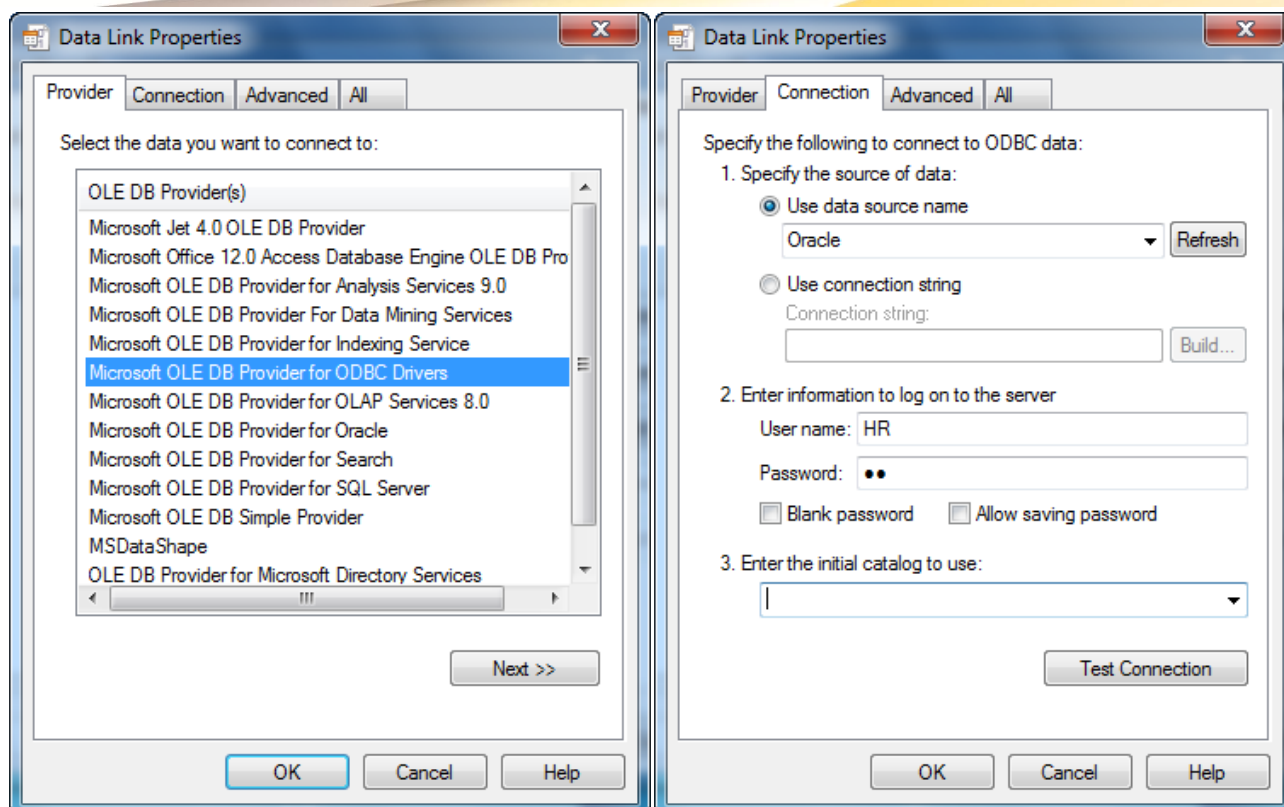


## Connecting to ODBC data source in Navicat

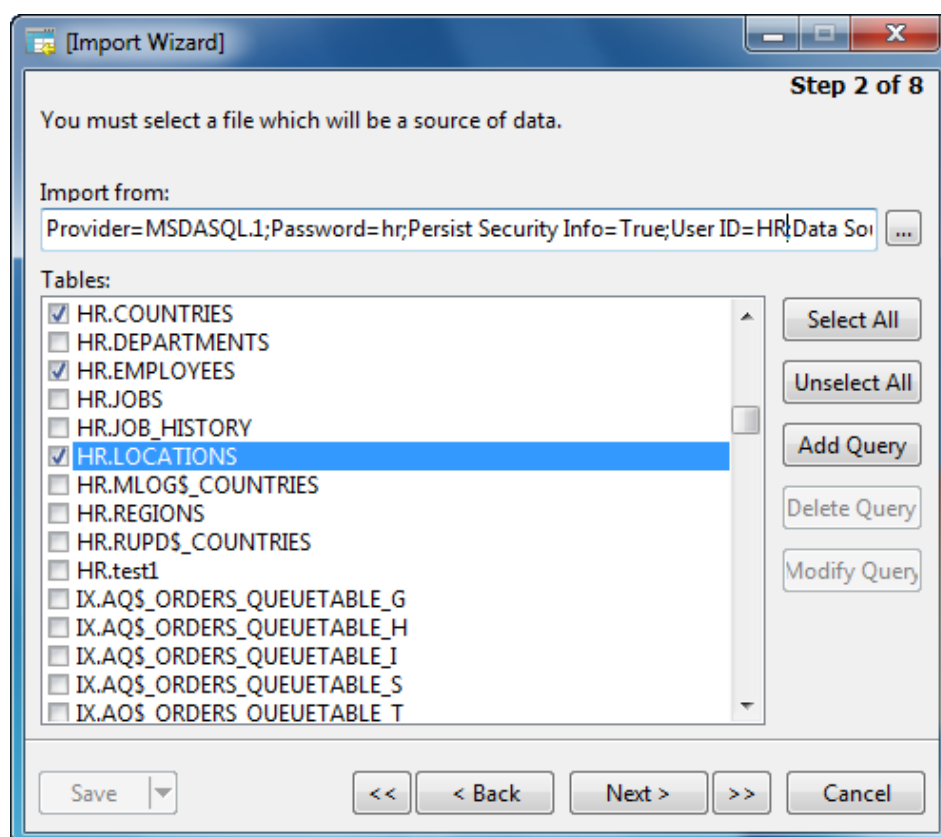
1. Click the **Import from** button in step 2 of the Import Wizard.



2. Under **Provider** tab in the **Data Link Properties**, select **Microsoft OLE DB Provider for ODBC Drivers**.  
Under **Connection** tab, choose the data source from the **Use data source name** drop-down list and provide valid username and password.



3. All available tables will be included in the list if connection success. Just simply choose the tables you wish to import or specify a query using **Add Query** button.

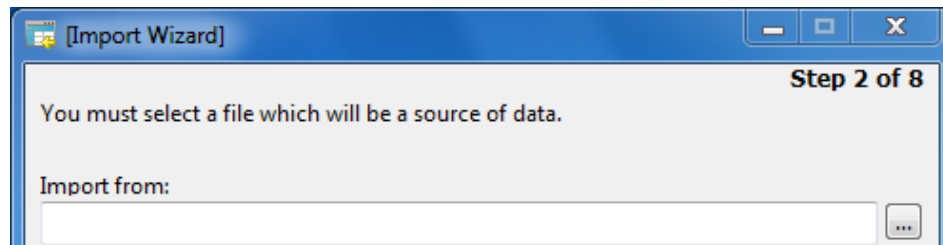




## Importing MSSQL Data (Step 2)

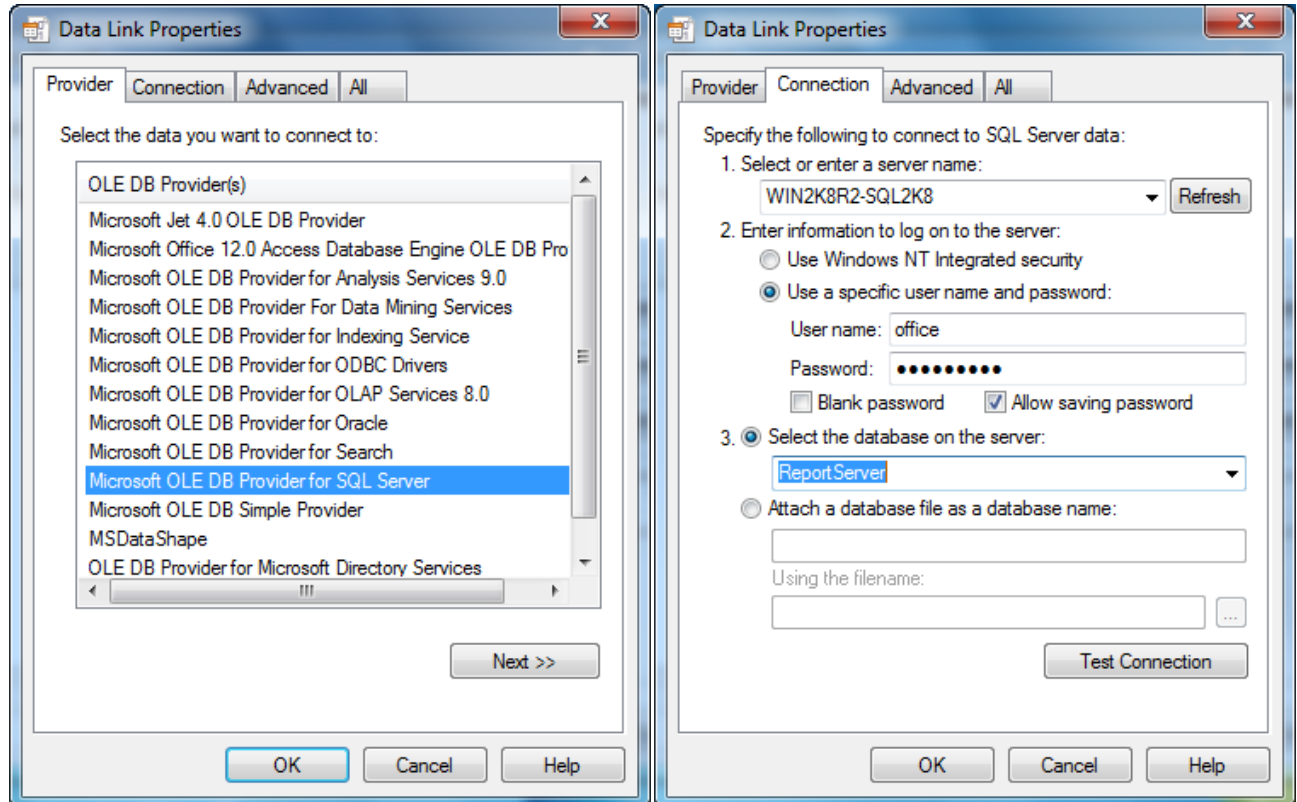
### Connecting to MSSQL

1. Click the **Import from** button in step 2 of the Import Wizard.

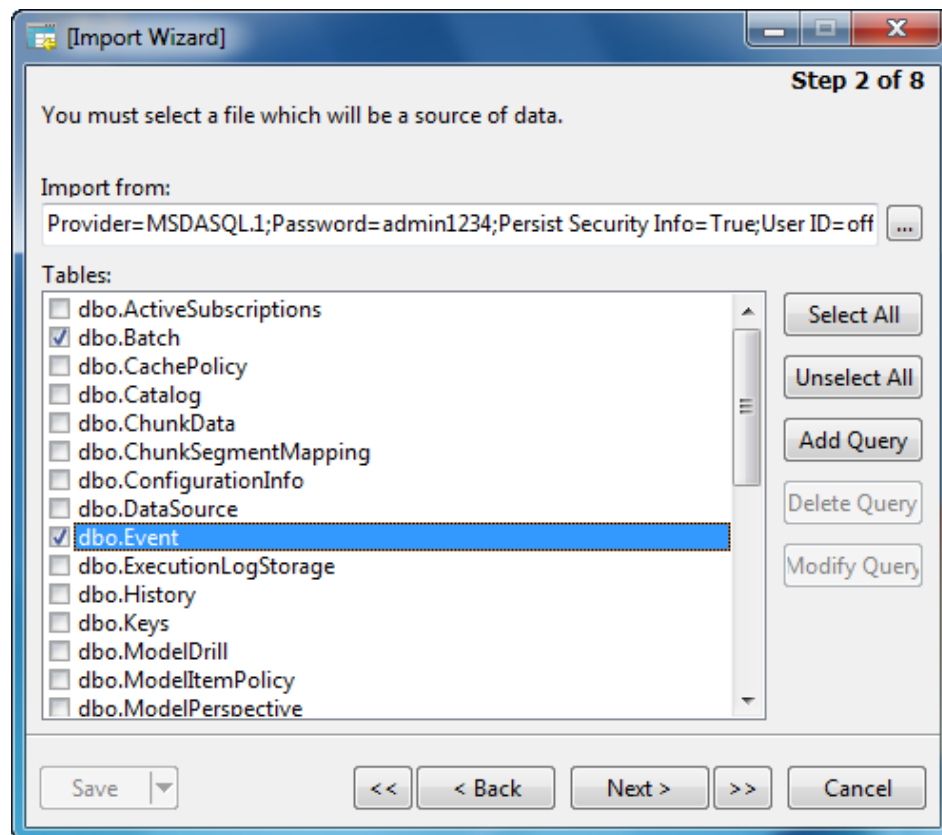


2. Under **Provider** tab in the **Data Link Properties**, select **Microsoft OLE DB Provider for SQL Server**.

Under **Connection** tab, choose the data source and database from the **Select or enter a server name** and **Select the database on the server** drop-down list respectively.



3. All available tables will be included in the list if connection success. Just simply choose the tables you wish to import or specify a query using **Add Query** button.



## Setting Additional Options for Specific File Type

Additional options specifies for file type.

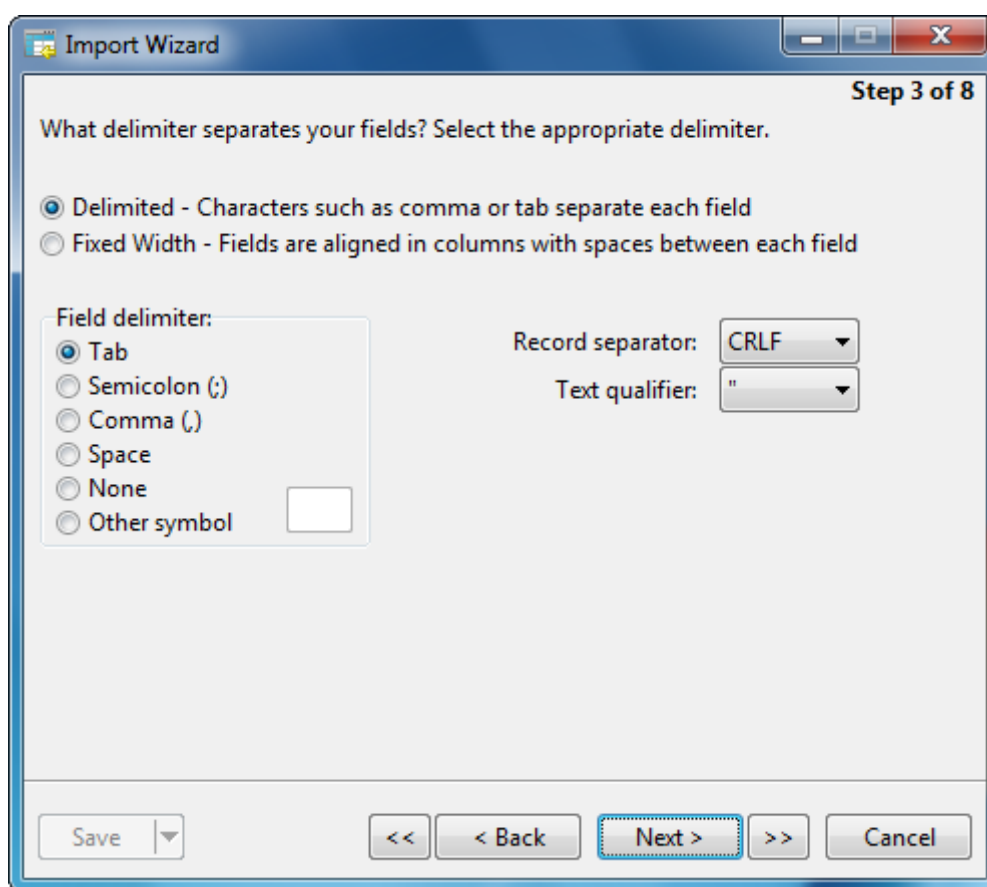
- [Setting Delimiter \(Step 3\) - TXT, XML](#)
- [Setting Data Format \(Step 4\) - TXT, XML, Excel, HTML](#)

## Setting Delimiter (Step 3) - TXT, XML

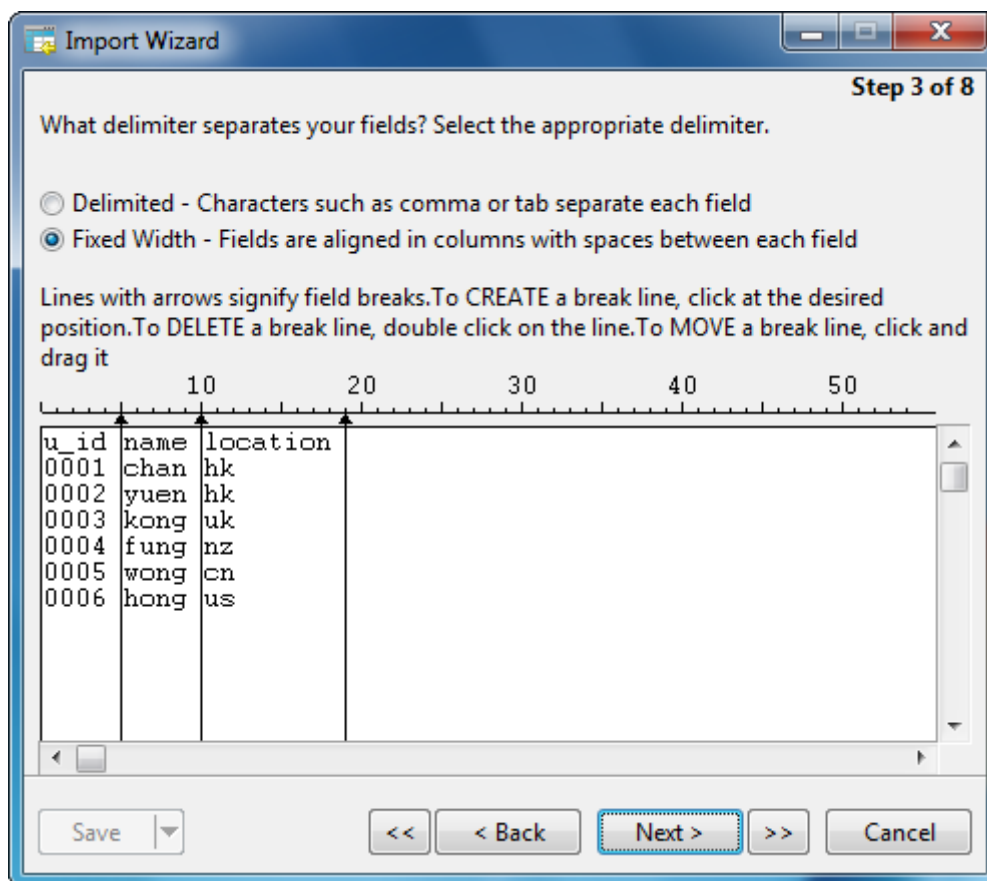
### TXT

Define **Field delimiter**, **Record separator** and **Text qualifier** for file. Record separator indicates how the file recognizes as new record (row).

**Note:** You should choose **Comma** for Field delimiter if you are importing CSV file.



Choose **Fixed Width** to import the text file with fixed width format. To delimit the source column bounds, click on the desired position. To remove it, just simply double-click the break line.



## XML

Define tag to identify table row.

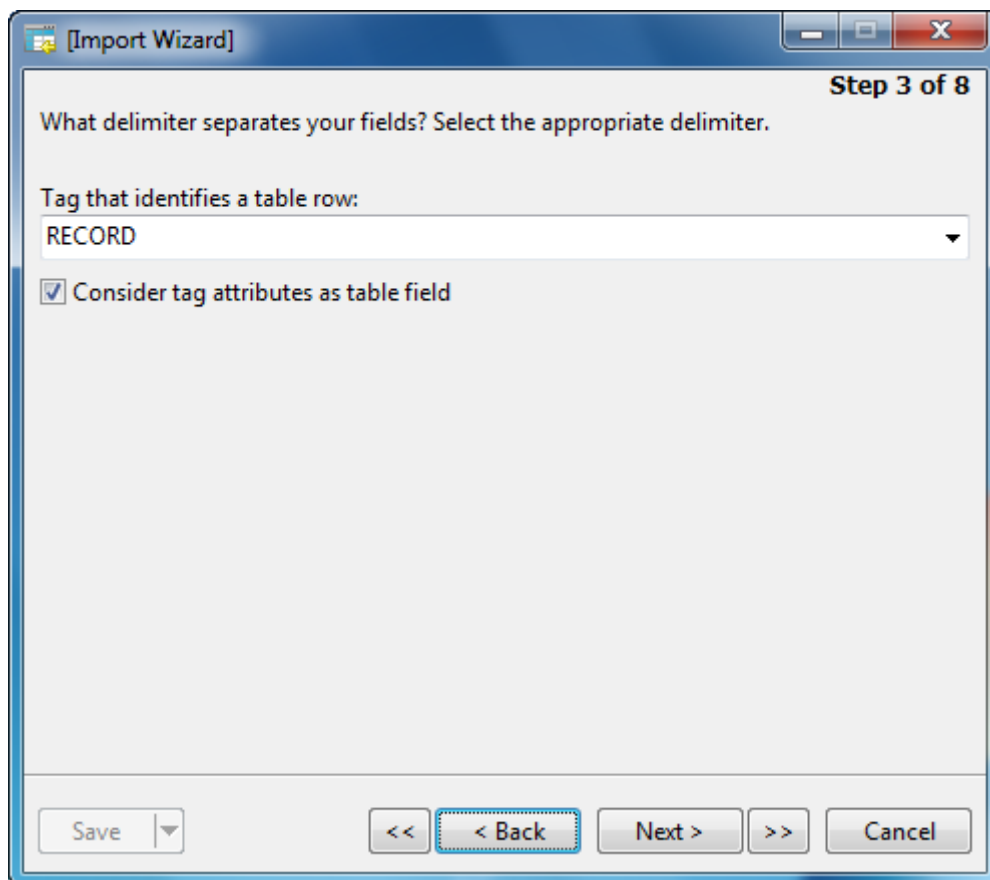
### ☒ **Consider tag attributes as table field**

For example:

```
<row age="17">  
<id>1</id>  
<name>sze</name>  
</row>
```

With this option is on, Navicat will recognizes "age" as a table field together with "id" and "name", otherwise, only "id" and "name" will be imported as table fields.

**Note:** Navicat does not support multiple level of XML file.



## Setting Data Format (Step 4) - TXT, XML, Excel, HTML

Import Wizard provides a number of options for setting common formats for all imported data.

### Field name row

Field name row indicates which row should Navicat recognize as Column Title.

### First data row

First data row indicates which row should Navicat start reading the actual data.

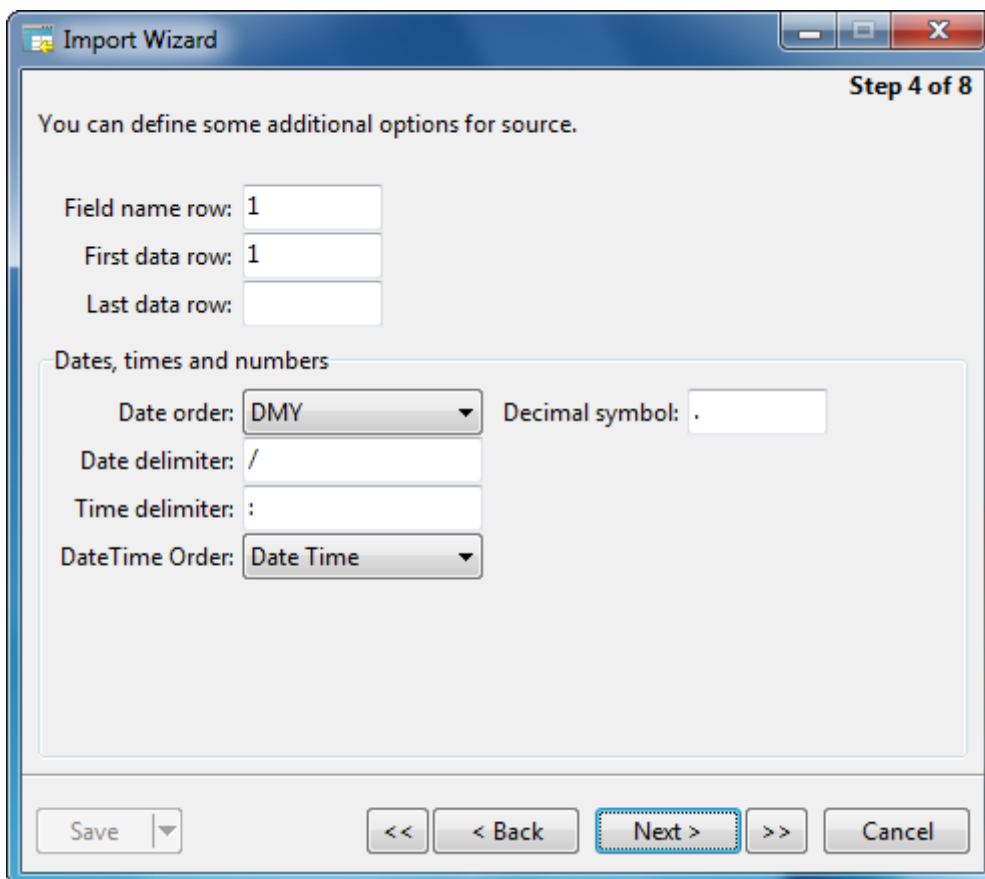
### Last data row

Last data row indicates which row should Navicat stop reading the actual data.

**Note:** If no column title are defined for the file, please enter **1** for First data row and **0** for Field name row.

### Dates, times and numbers

Defines the formats of the date, time and number.



Import Wizard

Step 4 of 8

You can define some additional options for source.

Field name row: 1

First data row: 1

Last data row:

Dates, times and numbers

Date order: DMY

Date delimiter: /

Time delimiter: :

Decimal symbol: .

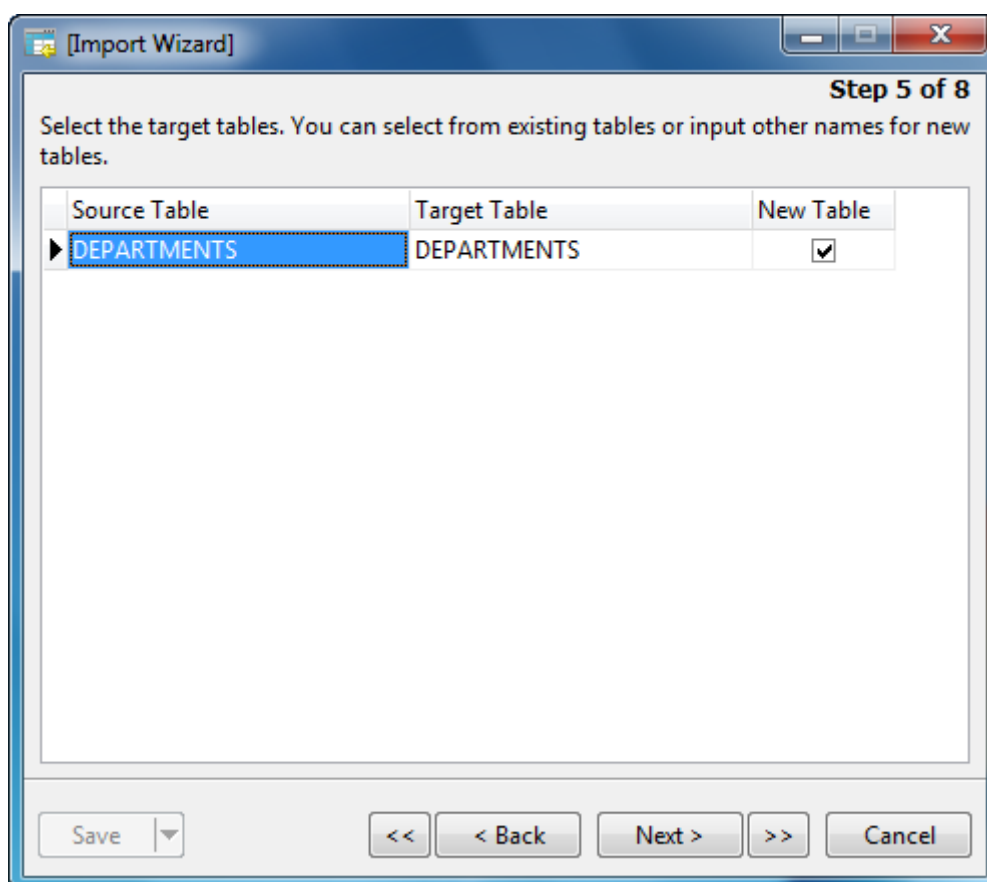
DateTime Order: Date Time

Save << < Back Next > >> Cancel

## Setting Target Table (Step 5)

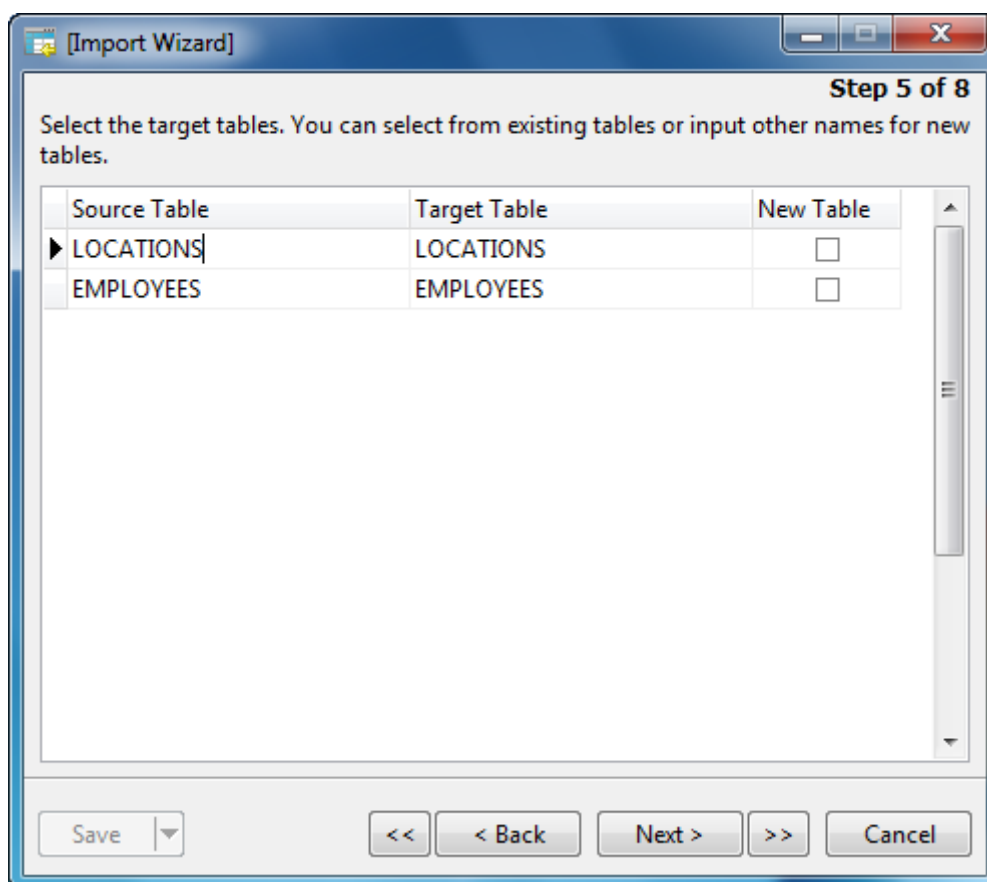
You are allowed to define a new table name or choose to import into the existing table from the drop-down list.

**Note:** If you type a new table name in **Target Table**, the box in **New Table** will be checked automatically.





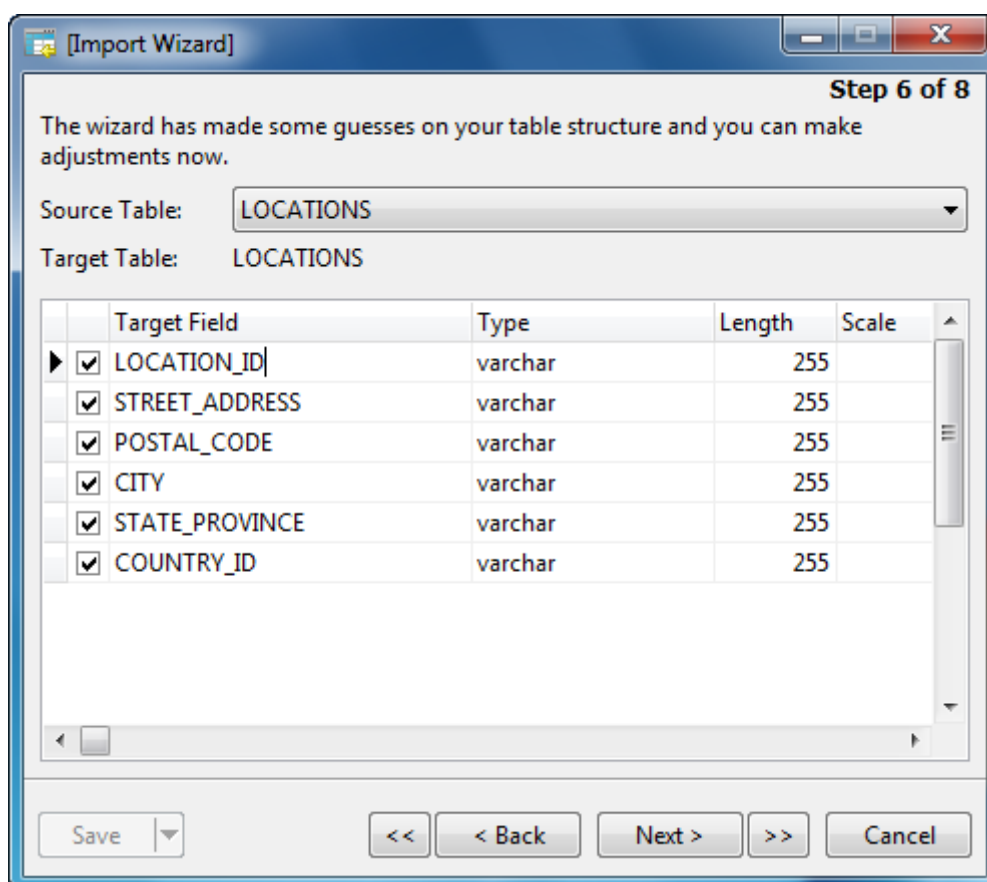
For importing multiple tables, all tables will be shown in the list.



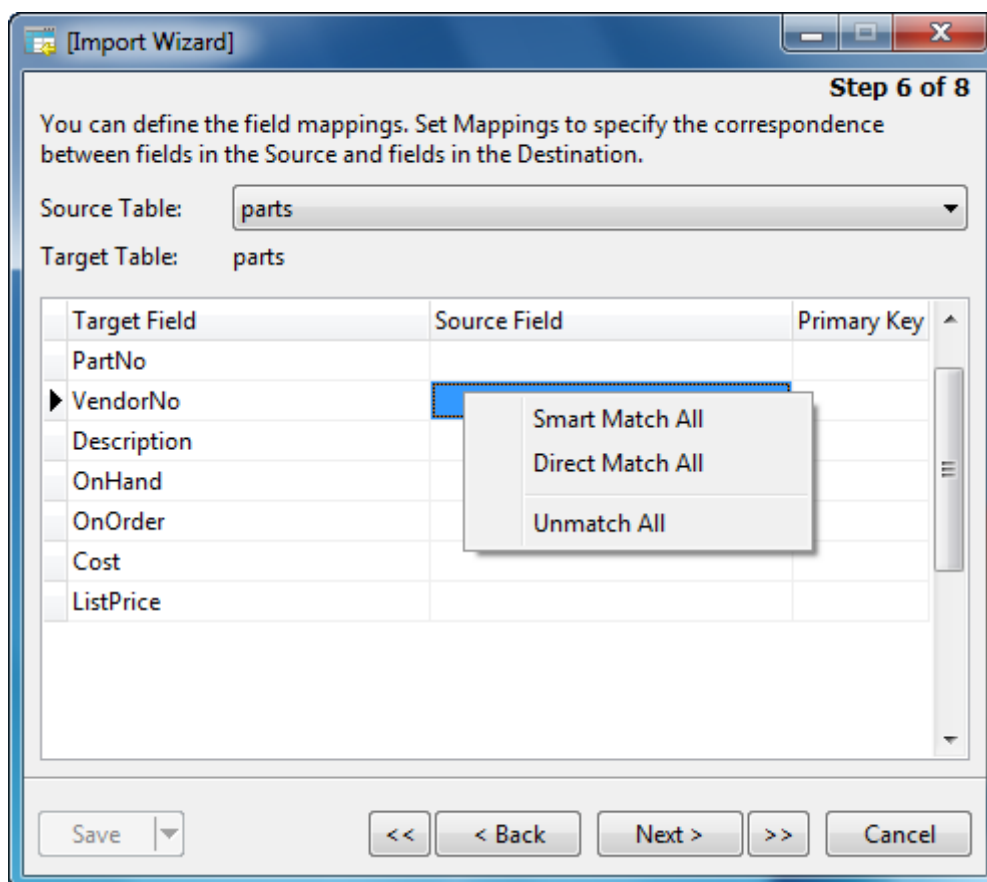
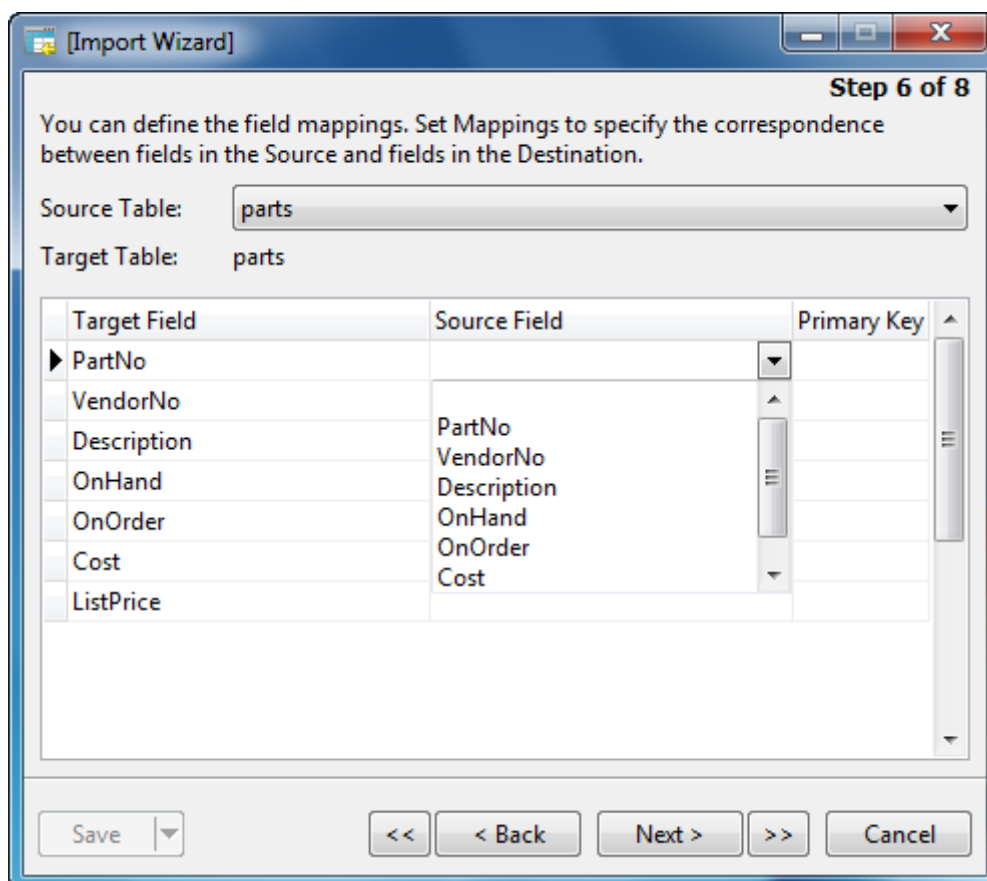
## Adjusting Field Structures and Mapping Fields (Step 6)

Navicat will make assumption on the field types and length in the source table. You are allowed to choose your desired type from the drop-down list.

**Hint:** For importing multiple tables, select the other tables from the **Source Table** drop-down list.

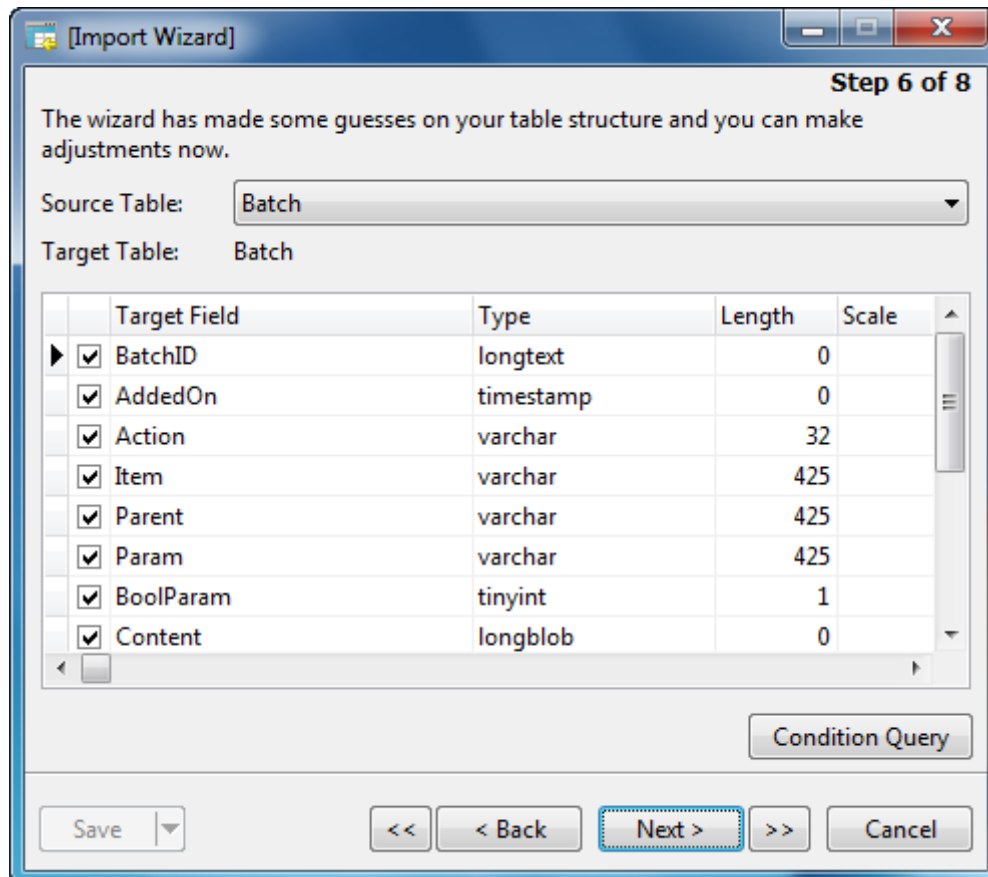


If you are importing your data into the existing table, then you might need to map the source field names manually to the destination table or just simply right-click and select **Smart Match All**, **Direct Match All** and **Unmatch All** from the popup menu for quick mapping.



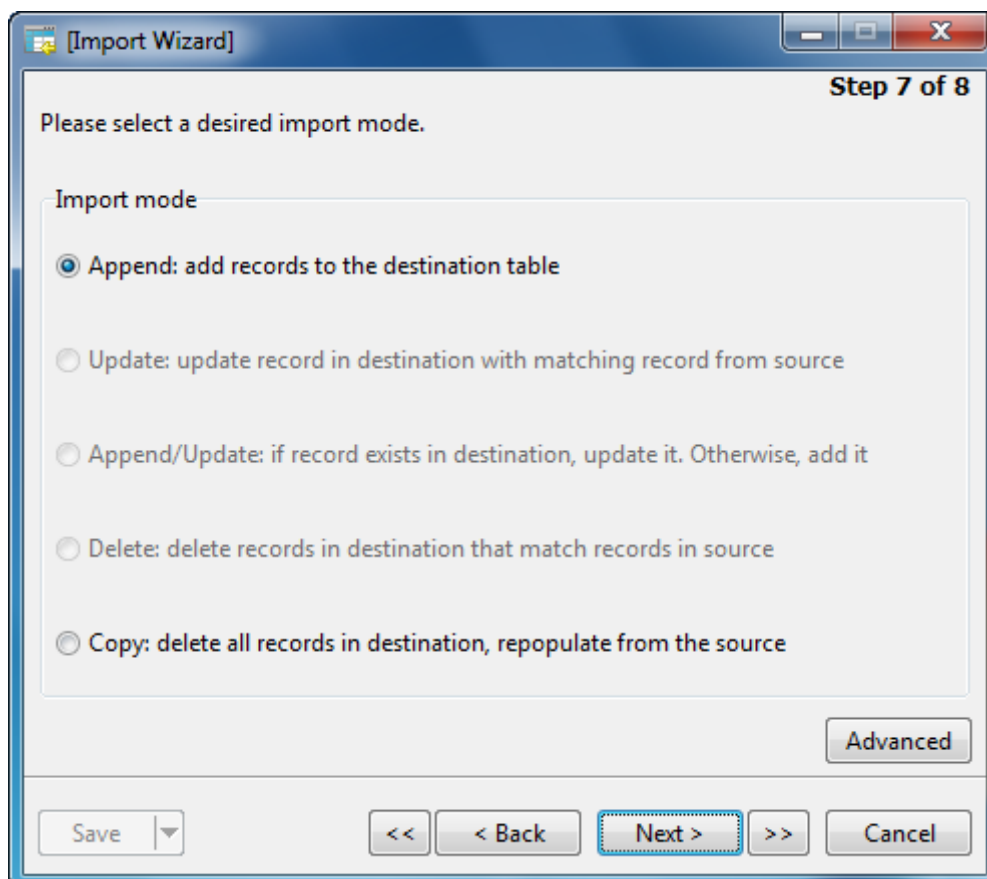
If you are importing via ODBC, the **Condition Query** button opens the **WHERE** dialog where you can specify a *WHERE* clause to import only certain rows from your source tables. In other words, just import only rows that satisfy the criteria set by you.

**Hint:** Do not include the word *WHERE* in the clause.



## Selecting Import Mode (Step 7)

Select the import mode that define how the data being imported.



**Hint:** To activate the remaining options, you must enable Primary Key in step 6.

	Target Field	Type	Length	Scale	Primary Key
<input checked="" type="checkbox"/>	LOCATION_ID	double	0	0	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	STREET_ADDRESS	varchar	120	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	POSTAL_CODE	varchar	36	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	CITY	varchar	90	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	STATE_PROVINCE	varchar	75	0	<input type="checkbox"/>
<input checked="" type="checkbox"/>	COUNTRY_ID	varchar	6	0	<input type="checkbox"/>

## Advanced

### ☒ **Run multiple queries in each execution (Available only for PostgreSQL and SQL Server)**

Checks this option if you want to run multiple queries in each execution.

### ☒ **Use extended insert statements (Available only for MySQL)**

Inserts records using extended insert syntax.

Example:

```
INSERT INTO `users` VALUES ('1', 'Peter McKindy', '23'), ('2', 'Johnson Ryne', '56'), ('0', 'Katherine', '23');
```

### ☒ **Use empty string as NULL**

Imports **NULL** value if the source data field contains empty string.

### ☒ **Use Foreign Key constraint (Available only for MySQL)**

Adds foreign key if there is foreign key relations between tables.

### ☒ **Continue on error**

Ignores errors that are encountered during the import process.

### ☒ **Include Unique, Index and Foreign Key**

Includes Unique, Index and foreign key during the import process.

**Note:** Support only when file type is MS Access database or ODBC.

### ☒ **Create Auto Increment Fields (Available only for MySQL and PostgreSQL)**

Creates Auto Increment Fields during the import process.

**Note:** Support only when file type is MS Access database, Paradox file or DBase file.

### ☒ **Import Deleted Records**

Import the deleted records in the DBase file during the import process.

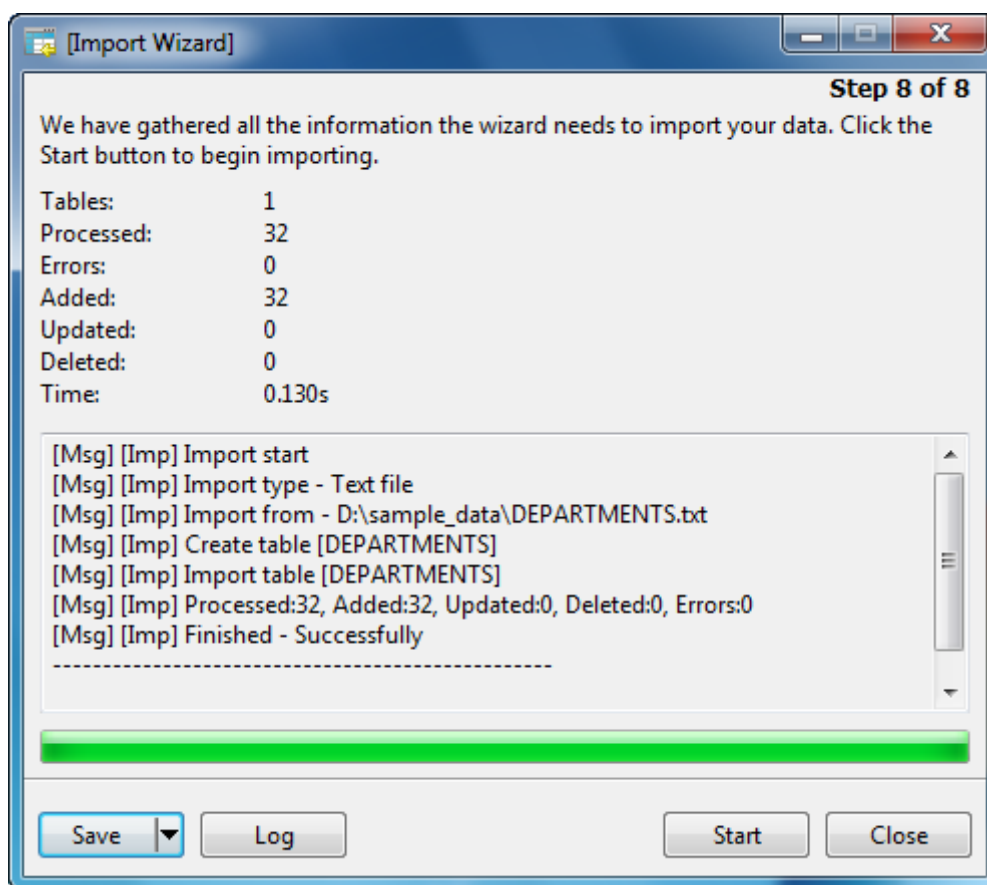
**Note:** Support only when file type is DBase file.

## Saving and Confirming Import (Step 8)

Click **Start** button to start the import process.

**Hint:** Click **Save** button to save your settings as a profile for setting schedule.

You can click **Log** button to view the running process indicating success or failure. These messages are saved in file - LogImport.txt.



## Export Wizard

**Export Wizard** allows you to export data from table, view, or query result to any available format. You can save your settings as a profile for setting schedule.

**Note:** Navicat Essentials version supports to export text-based files, such as TXT, CSV, HTML and XML file.

To open the Export Wizard, click  **Export Wizard** from the table object pane toolbar.

- [Settings Export File Format \(Step 1\)](#)
- [Setting Destination File Name and Encoding \(Step 2\)](#)
- [Selecting Fields for Export \(Step 3\)](#)
- [Setting Data Format \(Step 4\)](#)
- [Saving and Confirming Export \(Step 5\)](#)

To run a saved export profile from the command line

- Create and save the export profile.
- Start Navicat from command line, type the command (see [Command](#) for details)

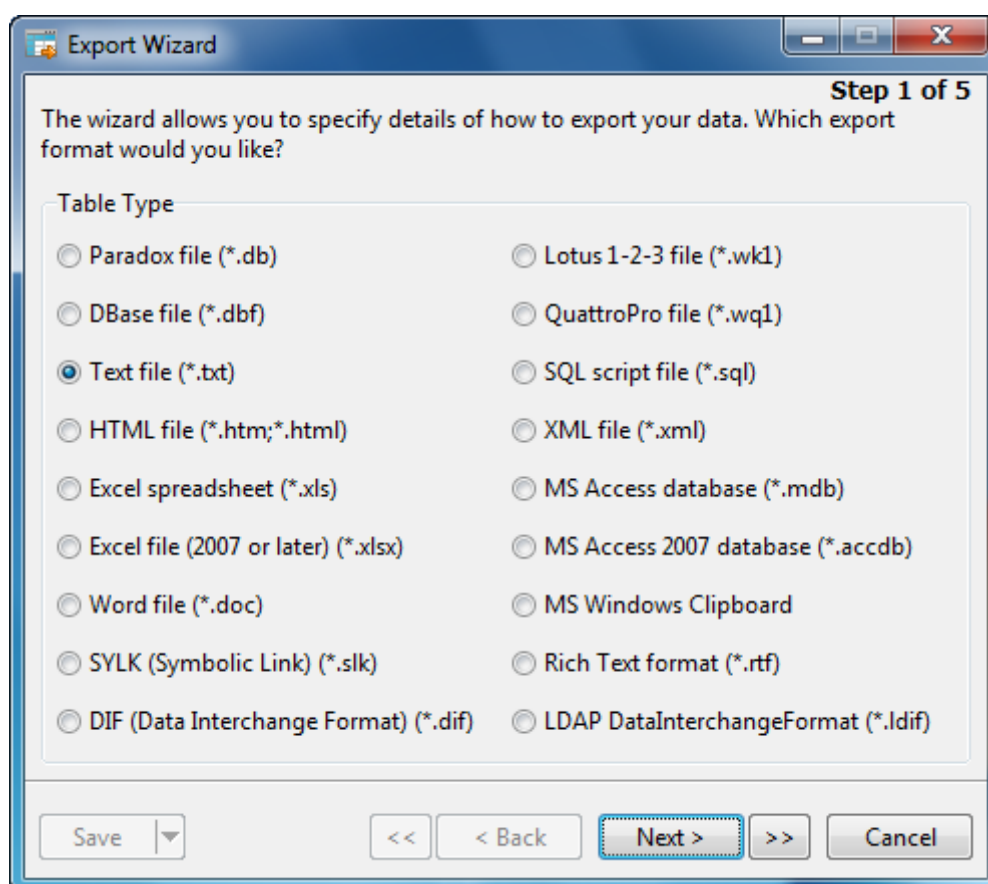


## Setting Export File Format (Step 1)

Select one of the available table formats.

**Note:** Navicat Essentials only supports exporting to TXT, CSV, HTML and XML file.

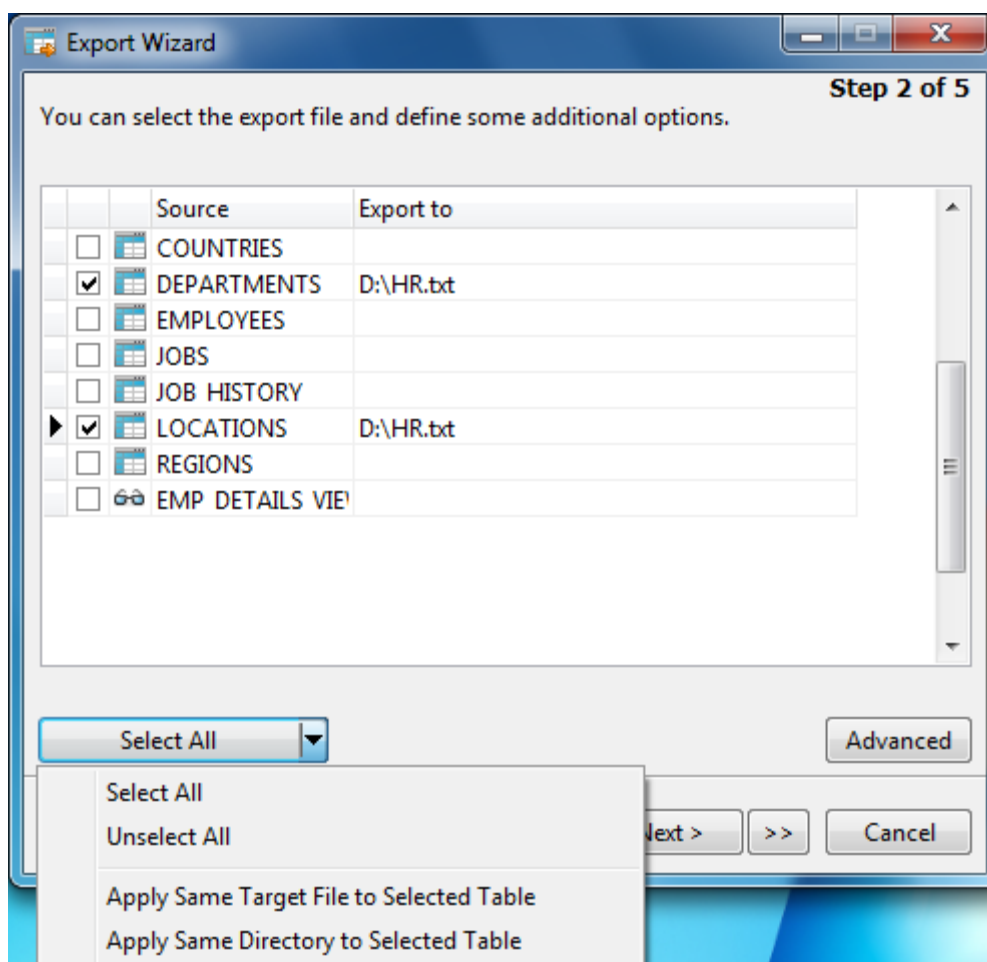
**Note:** The Excel file format is according to the Microsoft Office version installed in your computer.



## Setting Destination File Name and Encoding (Step 2)

Set name for the result file. The file name extension in the **Export to** text box changes according to the selected table type in step 1.

**Note:** For exporting query result, please ensure that you have saved the query before running the Export Wizard. Otherwise, no source table displayed in here.



## Select All

In Vista or above, you can select/unselect all exported tables by simply right-click and select **Select All** or **Unselect All** from the popup menu or from **Select All** button for quick mapping.

If you are exporting selected tables into the same target file, you can just simply right-click and select **Apply Same Target File to Selected Table** from the popup menu or from **Select All** button for quick mapping.

If you are exporting selected tables into the same directory, you can just simply right-click and select **Apply Same Directory to Selected Table** from the popup menu or from **Select All** button for quick mapping.

## Advanced

### Encoding

Select the encoding for the exported file.

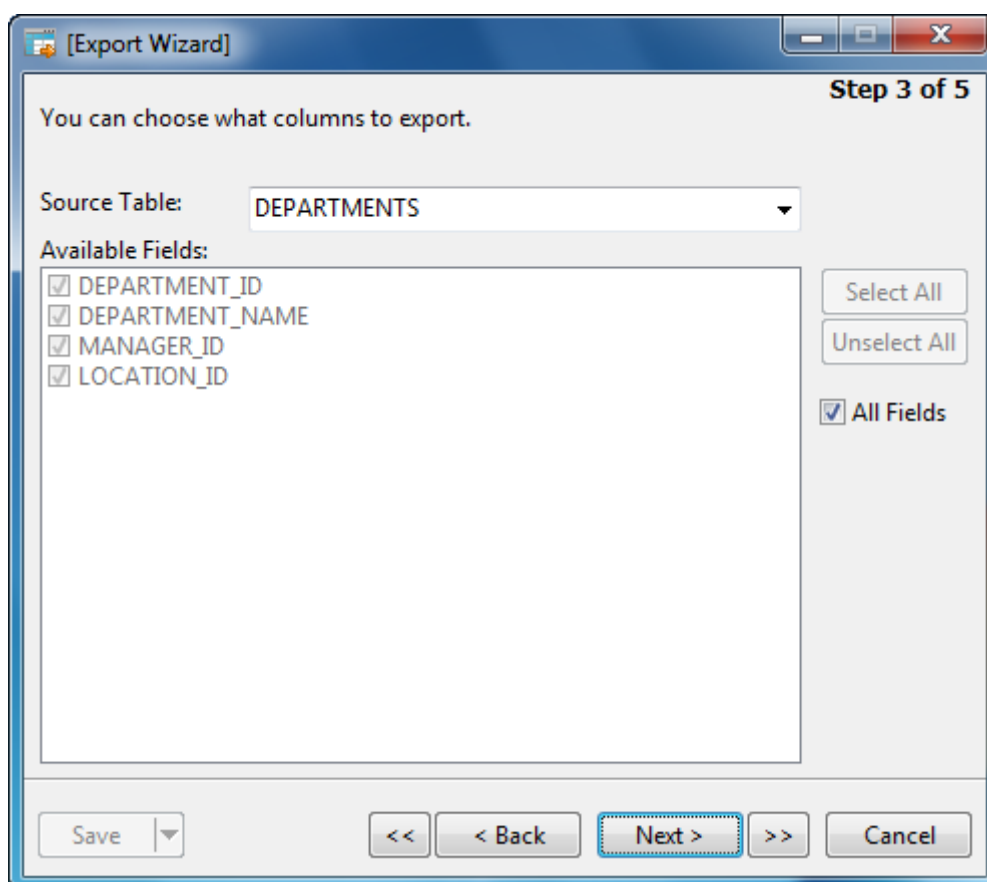
### ☒ Add timestamp

Checks this option if you want your file name specifies the timestamp of the export is run. Select the date/time format from the drop-down list.

## Selecting Fields for Export (Step 3)

Select table fields for export. All the fields are selected in the **Available Fields** list by default. If you want to omit some fields to be exported, just simply uncheck the box **All Fields** first and then uncheck those fields in the Available Fields list.

**Note:** For exporting query result, the wizard will skip this step.



## Setting Data Format (Step 4)

You are allowed to customize formats applied to exported data.

### ☒ **Include column titles**

Field names will be included into the exported file if this option is on.

### ☒ **Append**

Appends records to the existing file. If you select **Apply Same Target File to Selected Table** option for multiple tables in step 2, checks this option to append the records.

### ☒ **Continue on error**

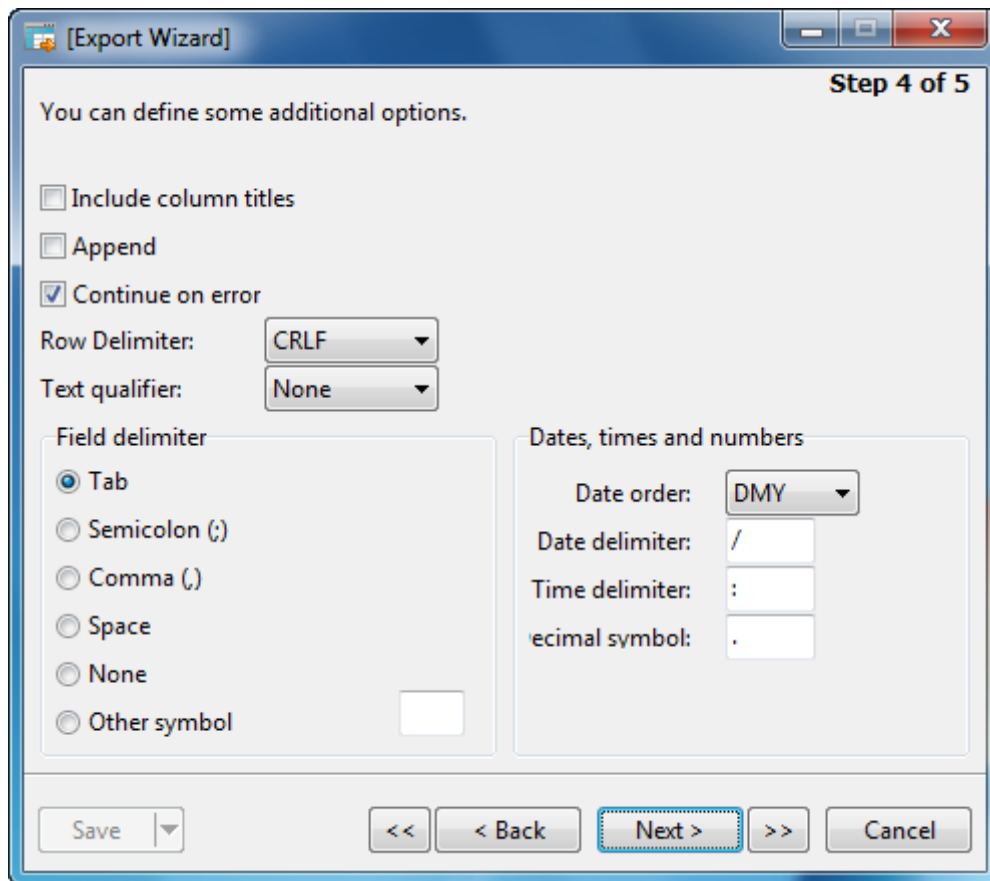
Ignores errors that are encountered during the export process.

### ☒ **Use Attributes Format in XML** (apply on XML format only)

Attributes Format
<pre>&lt;RECORDS&gt; &lt;RECORD OrderNo="1003" ItemNo="1" PartNo="1313" Qty="5" Discount="0"&gt;&lt;/RECORD&gt; &lt;RECORD OrderNo="1004" ItemNo="1" PartNo="1313" Qty="10" Discount="50"&gt;&lt;/RECORD&gt; &lt;/RECORDS&gt;</pre>
Non-Attributes Format
<pre>&lt;RECORDS&gt; &lt;RECORD&gt;   &lt;OrderNo&gt;1003&lt;/OrderNo&gt;   &lt;ItemNo&gt;1&lt;/ItemNo&gt;   &lt;PartNo&gt;1313&lt;/PartNo&gt;   &lt;Qty&gt;5&lt;/Qty&gt;   &lt;Discount&gt;0&lt;/Discount&gt; &lt;/RECORD&gt; &lt;RECORD&gt;   &lt;OrderNo&gt;1004&lt;/OrderNo&gt;   &lt;ItemNo&gt;1&lt;/ItemNo&gt;   &lt;PartNo&gt;1313&lt;/PartNo&gt;   &lt;Qty&gt;10&lt;/Qty&gt;   &lt;Discount&gt;50&lt;/Discount&gt; &lt;/RECORD&gt;</pre>

</RECORDS>

**Hint:** Only related options will be enabled according to the selected table type in step 1.



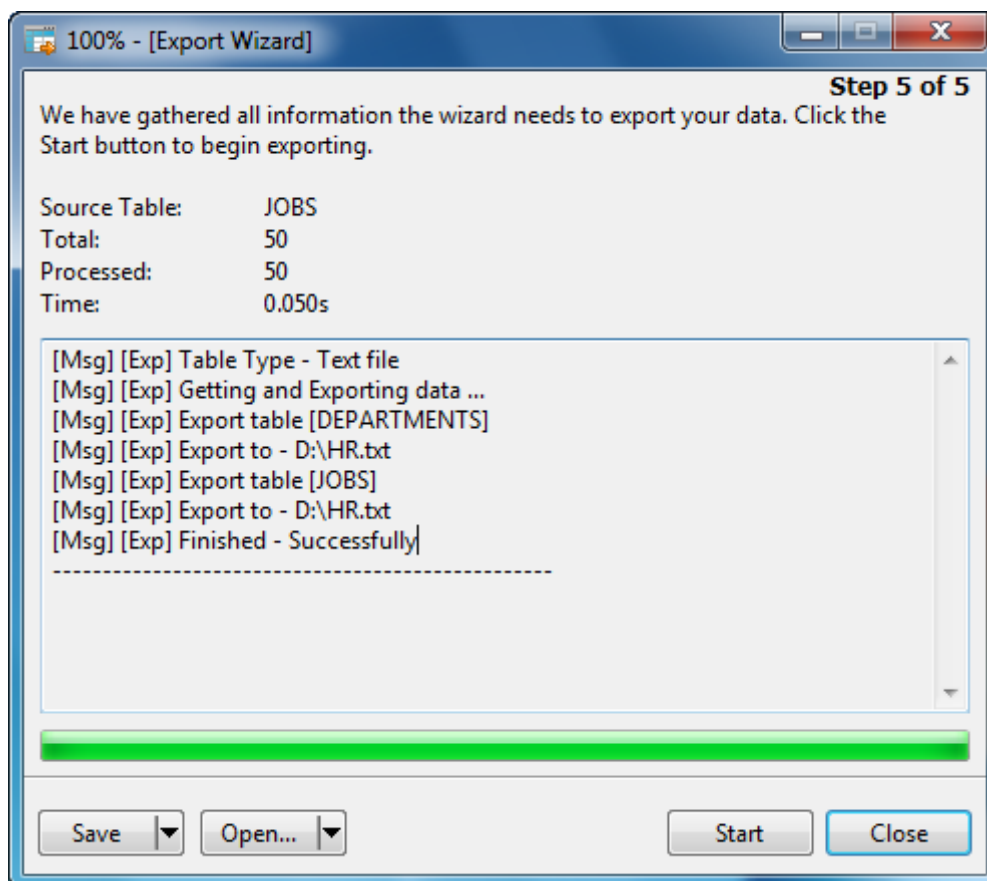
## Saving and Confirming Export (Step 5)

Click **Start** button to start the export process.

**Hint:** Click **Save** button to save your settings as a profile for setting schedule.

You can view the running process indicating success or failure. These messages are saved in file - LogExport.txt.

Click the **Open** button to open the log file or the exported file.





## Data Transfer (Available only in Full Version)

Navicat allows you to transfer tables/views/functions/sequences/events from one database/schema to another database/schema, or to a sql file. The target database/schema can be on the same server as the source database/schema or on another server. It also allows you to save a profile for easy retrieval and running of data transfer between databases/schemas. You can also invoke data transfer from the command line, which makes it possible to schedule data transfer between databases/schemas. You can save your settings as a profile for setting schedule.



Simply open the data transfer and use the data transfer toolbar, which allows you to create, save and delete the data transfer.


### Create Data Transfer

To create a new data transfer

- Select **Tools** ->  **Data Transfer...** from the main menu or just select  **New** from the toolbar above.
- Edit data transfer properties on the appropriate tabs.


To create a new data transfer with modification as one of the existing data transfer profiles

- Select **Tools** ->  **Data Transfer...** from the main menu
- Select the data transfer for modifying from the drop-down list.
- Modify data transfer properties on the appropriate tabs.
- Click  **Save As.**

**Hint:** To create new data transfer, you can also right-click the Database node in the navigation pane and select the  **Data Transfer...** from the popup menu.

### Edit Data Transfer

To edit the existing data transfer

- Select **Tools** ->  **Data Transfer...** from the main menu.
- Select the data transfer for modifying from the drop-down list.
- Modify data transfer properties on the appropriate tabs.



## Run Data Transfer

To run a data transfer



- Create a new data transfer/open the existing one.
- Click **Start**.

To run a saved data transfer profile from the command line

- Create and save the data transfer profile.
- Start Navicat from command line, type the command (see [Command](#) for details)

## Delete Data Transfer

To delete a data transfer

- Select **Tools** ->  **Data Transfer...** from the main menu.
- Select the data transfer from the drop-down list.
- Click the  **Delete** from the toolbar.
- Confirm deleting in the dialog window.

## General Settings for Data Transfer


The following instruction guides you through the process of setting up a data transfer. Customize options according to your needs. See drag and drop (MySQL, Oracle, PostgreSQL, SQLite or SQL Server).

### Source

Defines connection, database and schema for the source.

All the objects are selected in the **Database Objects** list by default. If you do not want some objects to be transferred, just simply uncheck them.

☒ With this option is on, only the checked objects will be transferred. However, if you add any new objects in the source database/schema after you create your data transfer profile, the newly added objects will not be transferred unless you manually modify the **Database Objects** list.

 Chooses this option if you wish all the objects being transferred to the target database/schema, all newly added objects will also be transferred without amending the data transfer profile.

### Target

#### **Connection**

Transfers your selected objects directly to the other database/schema. Chooses the connection and database/schema you wish to transfer to.

#### **File**

Transfers your selected objects directly to a text file. You can select different **SQL Format** and **Encoding** for the file.

## Advanced Settings for Same Server Type Data Transfer

### Table Options

☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

☒ **Include indexes**

Includes indexes in the table with this option is on.

☒ **Include foreign key constraints**

Includes foreign keys in the table with this option is on.

☒ **Include engine/table type (Available only for MySQL)**

Includes table type with this option is on.

☒ **Include character set (Available only for MySQL)**

Includes character set in the table with this option is on.

☒ **Include auto increment (Available only for MySQL, SQLite and SQL Server)**

Includes auto increment in the table with this option is on.

☒ **Include other table options (Available only for MySQL)**

Includes other options in the table with this option is on.

☒ **Include unique constraints (Available only for Oracle, PostgreSQL, SQLite and SQL Server)**

Includes uniques in the table with this option is on.

☒ **Include rules (Available only for PostgreSQL)**

Includes rules in the table with this option is on.

☒ **Include check constraints (Available only for Oracle, PostgreSQL, SQLite and SQL Server)**

Includes checks in the table with this option is on.

## ☒ **Include triggers**

Includes triggers in the table with this option is on.

## ☒ **Include excludes (Available only for PostgreSQL)**

Includes exclusion constraints in the table with this option is on.

## **Record Options**

### ☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

### ☒ **Lock target tables (Available only for MySQL, PostgreSQL and SQL Server)**

Locks the tables in the target database/schema during the data transfer process.

### ☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

### ☒ **Use complete insert statements (Available only for MySQL, Oracle, PostgreSQL and SQLite)**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

### ☒ **Use extended insert statements (Available only for MySQL)**

Inserts records using extended insert syntax.

Example:

```
INSERT INTO `users` VALUES ('1', 'Peter McKindy', '23'), ('2', 'Johnson Ryne',  
'56'), ('0', 'Katherine', '23');
```

## ☒ **Use delayed insert statements (Available only for MySQL)**

Inserts records using *DELAYED* insert SQL statements.

Example:

```
INSERT DELAYED INTO `users` VALUES ('1', 'Peter McKindy', '23');  
INSERT DELAYED INTO `users` VALUES ('2', 'Johnson Ryne', '56');  
INSERT DELAYED INTO `users` VALUES ('0', 'katherine', '23');
```

## ☒ **Run multiple insert statements (Available only for PostgreSQL and SQL Server)**

Check this option if you want to run multiple insert statements in each execution, which will make the data transfer process faster.

## ☒ **Use hexadecimal format for BLOB (Available only for MySQL, PostgreSQL, SQLite and SQL Server)**

Inserts BLOB data as hexadecimal format.

### **Other Options**

#### ☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

#### ☒ **Lock source tables (Available only for MySQL, Oracle, PostgreSQL and SQL Server)**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

#### ☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

#### ☒ **Create target database/schema if not exist (Available only for MySQL, Oracle, PostgreSQL and SQL Server)**

Creates a new database/schema if the database/schema specified in target server does not exist.

#### ☒ **Use DDL from SHOW CREATE TABLE (Available only for MySQL)**

If this option is on, DDL will be used from show create table.

#### ☒ **Use DDL from sqlite\_master (Available only for SQLite)**

If this option is on, DDL will be used from the *SQLITE\_MASTER* table.

## Advanced Settings for Cross Server Data Transfer (Available only in Navicat Premium)

Navicat Premium supports transferring data across different server types, e.g. from MySQL to Oracle. The Data Transfer process can transfer tables to the target. While the target server is MySQL or SQLite, the process can transfer tables with primary key constraints. The following part shows the settings for different server types.

- [Data transfer from MySQL to Oracle](#)
- [Data transfer from MySQL to PostgreSQL](#)
- [Data transfer from MySQL to SQLite](#)
- [Data transfer from MySQL to SQL Server](#)
- [Data transfer from Oracle to MySQL](#)
- [Data transfer from Oracle to PostgreSQL](#)
- [Data transfer from Oracle to SQLite](#)
- [Data transfer from Oracle to SQL Server](#)
- [Data transfer from PostgreSQL to MySQL](#)
- [Data transfer from PostgreSQL to Oracle](#)
- [Data transfer from PostgreSQL to SQLite](#)
- [Data transfer from PostgreSQL to SQL Server](#)
- [Data transfer from SQLite to MySQL](#)
- [Data transfer from SQLite to Oracle](#)
- [Data transfer from SQLite to PostgreSQL](#)
- [Data transfer from SQLite to SQL Server](#)
- [Data transfer from SQL Server to MySQL](#)
- [Data transfer from SQL Server to Oracle](#)
- [Data transfer from SQL Server to PostgreSQL](#)
- [Data transfer from SQL Server to SQLite](#)

## Advanced Settings for Transferring from MySQL to Oracle

### Table Options

#### ☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

#### ☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

#### ☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

#### ☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (`ID Number`, `User Name`, `User Age`) VALUES ('1',  
'Peter McKindsy', '23');  
INSERT INTO `users` (`ID Number`, `User Name`, `User Age`) VALUES ('2',  
'Johnson Ryne', '56');  
INSERT INTO `users` (`ID Number`, `User Name`, `User Age`) VALUES ('0',  
'katherine', '23');
```

### Other Options

#### ☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

#### ☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.



## Advanced Settings for Transferring from MySQL to PostgreSQL

### Table Options

#### ☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

#### ☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

#### ☒ **Lock target tables**

Locks the tables in the target database/schema during the data transfer process.

#### ☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

#### ☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (`ID Number`, `User Name`, `User Age`) VALUES ('1',  
'Peter McKindy', '23');
```

```
INSERT INTO `users` (`ID Number`, `User Name`, `User Age`) VALUES ('2',  
'Johnson Ryne', '56');
```

```
INSERT INTO `users` (`ID Number`, `User Name`, `User Age`) VALUES ('0',  
'katherine', '23');
```

#### ☒ **Run multiple insert statements**

Check this option if you want to run multiple insert statements in each execution, which will make the data transfer process faster.

#### ☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

## Other Options

☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.

## Advanced Settings for Transferring from MySQL to SQLite

### Table Options

#### ☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

#### ☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

#### ☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

#### ☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

#### ☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

## Other Options

### ☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

### ☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

## Advanced Settings for Transferring from MySQL to SQL Server

### Table Options

☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

☒ **Lock target tables**

Locks the tables in the target database/schema during the data transfer process.

☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

☒ **Run multiple insert statements**

Check this option if you want to run multiple insert statements in each execution, which will make the data transfer process faster.

☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

### Other Options

☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.

## Advanced Settings for Transferring from Oracle to MySQL

### Table Options

#### ☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

#### ☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

#### ☒ **Lock target tables**

Locks the tables in the target database/schema during the data transfer process.

#### ☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

#### ☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

#### ☒ **Use extended insert statements**

Inserts records using extended insert syntax.

Example:

```
INSERT INTO `users` VALUES ('1', 'Peter McKindy', '23'), ('2', 'Johnson Ryne',  
'56'), ('0', 'Katherine', '23');
```

## ☒ **Use delayed insert statements**

Inserts records using *DELAYED* insert SQL statements.

Example:

```
INSERT DELAYED INTO `users` VALUES ('1', 'Peter McKindy', '23');  
INSERT DELAYED INTO `users` VALUES ('2', 'Johnson Ryne', '56');  
INSERT DELAYED INTO `users` VALUES ('0', 'katherine', '23');
```

## ☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

## **Other Options**

### ☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

### ☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

### ☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

### ☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.



## Advanced Settings for Transferring from Oracle to PostgreSQL

### Table Options

☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

☒ **Lock target tables**

Locks the tables in the target database/schema during the data transfer process.

☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

☒ **Run multiple insert statements**

Check this option if you want to run multiple insert statements in each execution, which will make the data transfer process faster.

☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

## Other Options

☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.

## Advanced Settings for Transferring from Oracle to SQLite

### Table Options

#### ☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

#### ☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

#### ☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

#### ☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

#### ☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

## Other Options

### ☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

### ☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

## Advanced Settings for Transferring from Oracle to SQL Server

### Table Options

☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

☒ **Lock target tables**

Locks the tables in the target database/schema during the data transfer process.

☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

☒ **Run multiple insert statements**

Check this option if you want to run multiple insert statements in each execution, which will make the data transfer process faster.

☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

### Other Options

☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.

## Advanced Settings for Transferring from PostgreSQL to MySQL

### Table Options

#### ☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

#### ☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

#### ☒ **Lock target tables**

Locks the tables in the target database/schema during the data transfer process.

#### ☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

#### ☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

#### ☒ **Use extended insert statements**

Inserts records using extended insert syntax.

Example:

```
INSERT INTO `users` VALUES ('1', 'Peter McKindy', '23'), ('2', 'Johnson Ryne',  
'56'), ('0', 'Katherine', '23');
```

## ☒ **Use delayed insert statements**

Inserts records using *DELAYED* insert SQL statements.

Example:

```
INSERT DELAYED INTO `users` VALUES ('1', 'Peter McKindy', '23');  
INSERT DELAYED INTO `users` VALUES ('2', 'Johnson Ryne', '56');  
INSERT DELAYED INTO `users` VALUES ('0', 'katherine', '23');
```

## ☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

## **Other Options**

### ☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

### ☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

### ☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

### ☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.



## Advanced Settings for Transferring from PostgreSQL to Oracle

### Table Options

#### ☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

#### ☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

#### ☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

#### ☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

### Other Options

#### ☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

#### ☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.

## Advanced Settings for Transferring from PostgreSQL to SQLite

### Table Options

#### ☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

#### ☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

#### ☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

#### ☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

#### ☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

## Other Options

### ☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

### ☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

## Advanced Settings for Transferring from PostgreSQL to SQL Server

### Table Options

☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

☒ **Lock target tables**

Locks the tables in the target database/schema during the data transfer process.

☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

☒ **Run multiple insert statements**

Check this option if you want to run multiple insert statements in each execution, which will make the data transfer process faster.

☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

### Other Options

☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.

## Advanced Settings for Transferring from SQLite to MySQL Database

### Table Options

#### ☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

#### ☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

#### ☒ **Lock target tables**

Locks the tables in the target database/schema during the data transfer process.

#### ☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

#### ☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

#### ☒ **Use extended insert statements**

Inserts records using extended insert syntax.

Example:

```
INSERT INTO `users` VALUES ('1', 'Peter McKindy', '23'), ('2', 'Johnson Ryne',  
'56'), ('0', 'Katherine', '23');
```

## ☒ **Use delayed insert statements**

Inserts records using *DELAYED* insert SQL statements.

Example:

```
INSERT DELAYED INTO `users` VALUES ('1', 'Peter McKindy', '23');  
INSERT DELAYED INTO `users` VALUES ('2', 'Johnson Ryne', '56');  
INSERT DELAYED INTO `users` VALUES ('0', 'katherine', '23');
```

## ☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

## **Other Options**

### ☒ **Continue on errors**

Ignores errors that are encountered during the transfer process.

### ☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

### ☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

### ☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.



## Advanced Settings for Transferring from SQLite to Oracle

### Table Options

☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

### Other Options

☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.

## Advanced Settings for Transferring from SQLite to PostgreSQL

### Table Options

#### ☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

#### ☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

#### ☒ **Lock target tables**

Locks the tables in the target database/schema during the data transfer process.

#### ☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

#### ☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

#### ☒ **Run multiple insert statements**

Check this option if you want to run multiple insert statements in each execution, which will make the data transfer process faster.

#### ☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

## Other Options

☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.

## Advanced Settings for Transferring from SQLite to SQL Server

### Table Options

☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

☒ **Lock target tables**

Locks the tables in the target database/schema during the data transfer process.

☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

☒ **Run multiple insert statements**

Check this option if you want to run multiple insert statements in each execution, which will make the data transfer process faster.

☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

### Other Options

☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.

## Advanced Settings for Transferring from SQL Server to MySQL

### Table Options

#### ☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

#### ☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

#### ☒ **Lock target tables**

Locks the tables in the target database/schema during the data transfer process.

#### ☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

#### ☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

#### ☒ **Use extended insert statements (Available only for MySQL)**

Inserts records using extended insert syntax.

Example:

```
INSERT INTO `users` VALUES ('1', 'Peter McKindy', '23'), ('2', 'Johnson Ryne',  
'56'), ('0', 'Katherine', '23');
```

## ☒ **Use delayed insert statements (Available only for MySQL)**

Inserts records using *DELAYED* insert SQL statements.

Example:

```
INSERT DELAYED INTO `users` VALUES ('1', 'Peter McKindy', '23');  
INSERT DELAYED INTO `users` VALUES ('2', 'Johnson Ryne', '56');  
INSERT DELAYED INTO `users` VALUES ('0', 'katherine', '23');
```

## ☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

### **Other Options**

#### ☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

#### ☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

#### ☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

#### ☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.



## Advanced Settings for Transferring from SQL Server to Oracle

### Table Options

☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');  
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

### Other Options

☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.

## Advanced Settings for Transferring from SQL Server to PostgreSQL

### Table Options

☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

☒ **Lock target tables**

Locks the tables in the target database/schema during the data transfer process.

☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

☒ **Run multiple insert statements**

Check this option if you want to run multiple insert statements in each execution, which will make the data transfer process faster.

☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

## Other Options

☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

☒ **Lock source tables**

Locks the tables in the source database so that any update on the table is not allowed once the data transfer is triggered off.

☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

☒ **Create target database/schema if not exist**

Creates a new database/schema if the database/schema specified in target server does not exist.

## Advanced Settings for Transferring from SQL Server to SQLite

### Table Options

#### ☒ **Create tables**

Creates tables in the target database with this option is on.

Supposes this option is unchecked and tables already exist in the target database, then all data will be appended to the destination tables.

### Record Options

#### ☒ **Insert records**

Check this option if you require all records to be transferred to the destination database/schema.

#### ☒ **Use transaction**

Check this option if you use transaction during the data transfer process.

#### ☒ **Use complete insert statements**

Inserts records using complete insert syntax.

Example:

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('1',  
'Peter McKindy', '23');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('2',  
'Johnson Ryne', '56');
```

```
INSERT INTO `users` (` ID Number`, ` User Name`, ` User Age`) VALUES ('0',  
'katherine', '23');
```

#### ☒ **Use hexadecimal format for BLOB**

Inserts BLOB data as hexadecimal format.

## Other Options

### ☒ **Continue on error**

Ignores errors that are encountered during the transfer process.

### ☒ **Drop target objects before create**

Check this option if objects already exist in the target database/schema, the existing objects will be deleted once the data transfer starts.

## Data Transfer Message Log

The **Message Log** tab allows you to view the running process indicating success or failure.

Example:

```
[Msg] [Dtf] DataTransfer started
[Msg] [Dtf] Getting tables properties
[Msg] [Dtf] Getting tables fields
[Msg] [Dtf] Getting tables constraints
[Msg] [Dtf] Getting tables indexes
[Msg] [Dtf] Getting total records count
[Msg] [Dtf] Drop table: items
[Msg] [Dtf] Create table: items
[Msg] [Dtf] Get table data for: items
[Msg] [Dtf] Start transfer to Target Server: items
[Msg] [Dtf] Begin transaction on target server
[Msg] [Dtf] End transaction on target server
[Msg] [Dtf] Drop view: view_for_mysql1
[Msg] [Dtf] Create view: view_for_mysql1
[Msg] [Dtf] Drop procedure: procedure1
[Msg] [Dtf] Create procedure: procedure1
[Msg] [Dtf] Finished - Successfully
```

## Data Synchronization (Available only in Full Version)

Navicat allows you to transfer data from one database/schema to another database/schema with detailed analytical process. In other words, Navicat provides the ability for data in different databases/schemas to be kept up-to-date so that each repository contains the same information. The target database/schema can be on the same server as the source database/schema or on another server. You are not only authorized to rollback the transferring process, but also insert, delete and update records to the destination. You can also save your settings as a data synchronization profile for setting schedule. Same as Data Transfer, Data Synchronization can be invoked from the command line.

### Note:

For Oracle Database:



- BLOB, CLOB, NCLOB, LONG and LONG RAW data are skipped during the data synchronization process.
- TIMESTAMP primary key cannot synchronize (insert, update) with Database Link to 9i server.
- RAW primary key cannot synchronize (insert, update, delete) with Database Link to any server, without error.

**Note:** Only SQL Server 2005 or above supports data synchronization.

Just simply open the data synchronization and use the data synchronization toolbar, allowing you to create, save and delete the data synchronization.



## Create Data Synchronization

To create a new data synchronization

- Select **Tools** ->  **Data Synchronization...** from the main menu or just select  **New** from the toolbar above.
- Edit data synchronization properties on the appropriate tabs.




To create a new data synchronization with modification as one of the existing data synchronization profiles

- Select **Tools** ->  **Data Synchronization...** from the main menu
- Select the data synchronization for modifying from the drop-down list.
- Modify data synchronization properties on the appropriate tabs.
- Click  **Save As**.

## Edit Data Synchronization

To edit the existing data synchronization

- Select **Tools** ->  **Data Synchronization...** from the main menu.
- Select the data synchronization for modifying from the drop-down list.
- Modify data synchronization properties on the appropriate tabs.

## Preview Data Synchronization

To preview a data synchronization before execution

- Create a new data synchronization/open the existing one.
- Click **Preview**.

## Run Data Synchronization

To run a data synchronization



- Create a new data synchronization/open the existing one.
- Click **Start**.

To run a saved data synchronization profile from the command line

- Create and save the data synchronization profile.
- Start Navicat from command line, type the command (see Command for details)

## Delete Data Synchronization

To delete a data synchronization

- Select **Tools** ->  **Data Synchronization...** from the main menu.
- Select the data synchronization for dropping from the drop-down list.
- Click the  **Delete** from the toolbar.
- Confirm deleting in the dialog window.

## General Settings for Data Synchronization

The following instruction guides you through the process of setting up a data synchronization. Customize options according to your needs.

**Note:** All tables must contain primary key(s) and all table structures must be identical between the source and target (see Structure Synchronization).

### Source

Defines connection, database and schema for the source.

### Target

Defines connection, database and schema for the target.

**Note:** For Oracle server, you need to create Public/Private Database Link to the target Oracle server database before.

### Source Table/Target Table

Only tables which contain identical table names between the source and target are mapped in the list by default. If you do not want some tables to be synchronized, simply disable them manually from the drop-down list.

**Hint:** You can preview the outcome before execution.

## Advanced Settings for Data Synchronization

### ☒ **Use Transaction**

Rollbacks all data when error occurs.

### ☒ **Show synchronization detail**

Check this option if you want to list the details process under the message log tab during the synchronization.

**Note:** The process will be faster if this option is unchecked.

### ☒ **Insert records, Delete records, Update records**

Check these options to performing such actions to the target when data are synchronized.

## Data Synchronization Message Log

The **Message Log** tab allows you to view the running process indicating success or failure. These messages are saved in file - LogSynchronize.txt.

Example:

```
[Msg] [Dsy] Syn Start...  
[Msg] [Dsy] Synchronize table: localhost.report_sample.clients ->  
remote.report_sample.clients  
[Msg] [Dsy] total 5, equal 5, insert 0, update 0, delete 0  
[Msg] [Dsy] Time elapsed: 0.031s  
[Msg] [Dsy] Syn Success  
[Msg] [Dsy] Finished - Successfully
```



## Structure Synchronization (Available only in Full version & only for MySQL, Oracle, PostgreSQL and SQL Server)

Navicat allows you to compare and modify the table structures with detailed analytical process. In other words, Navicat compares tables between two databases/schemas and states the differential in structure. The target database/schema can be on the same server as the source database/schema or on another server.



Open the structure synchronization and use the structure synchronization toolbar, allowing you to create, save and delete the structure synchronization.

### Create Structure Synchronization

To create a new structure synchronization


- Select **Tools** ->  **Structure Synchronization...** from the main menu or just select  **New** from the toolbar above.
- Edit structure synchronization properties on the General tab.

To create a new structure synchronization with modification as one of the existing structure synchronization profiles

- Select **Tools** ->  **Structure Synchronization...** from the main menu
- Select the structure synchronization for modifying from the drop-down list.
- Modify structure synchronization properties on the General tab.
- Click  **Save As**.

### Edit Structure Synchronization

To edit the existing structure synchronization

- Select **Tools** ->  **Structure Synchronization...** from the main menu.
- Select the structure synchronization for modifying from the drop-down list.
- Modify structure synchronization properties on the General tab.



## Run Structure Synchronization

To run a structure synchronization

- Create a new structure synchronization/open the existing one.
- Click **Compare** to generate a set of scripts which shows the differentiation between the databases/schemas.
- Select the scripts you want to run.
- Click **Run Query**.

## Delete Structure Synchronization

To delete a structure synchronization

- Select **Tools** ->  **Structure Synchronization...** from the main menu.
- Select the structure synchronization for dropping from the drop-down list.
- Click the  **Delete** from the toolbar.
- Confirm deleting in the dialog window.

## General Settings for MySQL Structure Synchronization

The following instruction guides you through the process of setting up a structure synchronization. Customize options according to your needs.

### Source

Defines connection and database for the source.

### Target

Defines connection and database for the target.

### Compare Options

#### ☒ **Compare Tables**

Check this option if you want to compare tables between the source and target databases. Select/unselect the seven options below:

#### ☒ **Compare Primary Keys**

Check this option if you want to compare table primary keys.

#### ☒ **Compare Foreign Keys**

Check this option if you want to compare table foreign keys.

#### ☒ **Compare Indexes**

Check this option if you want to compare indexes.

#### ☒ **Compare Triggers**

Check this option if you want to compare triggers.

#### ☒ **Compare Character Set**

Check this option if you want to compare character set of the tables.

#### ☒ **Compare Auto Increment Value**

Check this option if you want to compare table auto increment values.

#### ☒ **Compare Partitions**

Check this option if you want to compare table partitions.

#### ☒ **Compare Views**

Check this option if you want to compare views.



## ☒ **Compare Functions**

Check this option if you want to compare functions.

## ☒ **Compare Events**

Check this option if you want to compare events.

## **Execution Options**

### ☒ **SQL for objects to be created**

Check this option to include all related SQL statements if new objects will be created in the target database.

### ☒ **SQL for objects to be changed**

Check this option to include all related SQL statements if objects will be changed in the target database.

### ☒ **SQL for objects to be dropped**

Check this option to include all related SQL statements if objects will be dropped from the target database.

### ☒ **Compare after execution**

Compares tables after the synchronization is executed.

### ☒ **Continue on error**

Ignores errors that are encountered during the synchronization process.

## General Settings for Oracle Structure Synchronization

The following instruction guides you through the process of setting up a structure synchronization. Customize options according to your needs.

### Source

Defines connection and schema for the source.

### Target

Defines connection and schema for the target.

### Compare Options

#### ☒ **Compare Tables**

Check this option if you want to compare tables between the source and target schemas. Select/unselect the four options below:

##### ☒ **Compare Primary Keys**

Check this option if you want to compare table primary keys.

##### ☒ **Compare Foreign Keys**

Check this option if you want to compare table foreign keys.

##### ☒ **Compare Uniques**

Check this option if you want to compare uniques.

##### ☒ **Compare Checks**

Check this option if you want to compare checks.

##### ☒ **Compare Views**

Check this option if you want to compare views.

##### ☒ **Compare Functions**

Check this option if you want to compare functions.

##### ☒ **Compare Indexes**

Check this option if you want to compare indexes.

##### ☒ **Compare Sequences**

Check this option if you want to compare sequences.

## ☒ **Compare Triggers**

Check this option if you want to compare triggers.

## ☒ **Compare Tablespace and Physical Attributes**

Check this option if you want to compare tablespace and physical attributes.

## **Execution Options**

### ☒ **SQL for objects to be created**

Check this option to include all related SQL statements if new objects will be created in the target schema.

### ☒ **SQL for objects to be changed**

Check this option to include all related SQL statements if objects will be changed in the target schema.

### ☒ **SQL for objects to be dropped**

Check this option to include all related SQL statements if objects will be dropped from the target schema.

### ☒ **Drop with CASCADE**

Check this option if you want to cascade to drop the dependent objects.

### ☒ **Compare after execution**

Compares tables after the synchronization is executed.

### ☒ **Continue on error**

Ignores errors that are encountered during the synchronization process.

## General Settings for PostgreSQL Structure Synchronization

The following instruction guides you through the process of setting up a structure synchronization. Customize options according to your needs.

### Source

Defines connection, database and schema for the source.

### Target

Defines connection, database and schema for the target.

### Compare Options

#### ☒ **Compare Tables**

Check this option if you want to compare tables between the source and target schemas. Select/unselect the five options below:

##### ☒ **Compare Primary Keys**

Check this option if you want to compare table primary keys.

##### ☒ **Compare Foreign Keys**

Check this option if you want to compare table foreign keys.

##### ☒ **Compare Uniques**

Check this option if you want to compare uniques.

##### ☒ **Compare Checks**

Check this option if you want to compare checks.

##### ☒ **Compare Excludes**

Check this option if you want to compare exclude constraints.

##### ☒ **Compare Views**

Check this option if you want to compare views.

##### ☒ **Compare Functions**

Check this option if you want to compare functions.

##### ☒ **Compare Indexes**

Check this option if you want to compare indexes.

## ☒ **Compare Sequences**

Check this option if you want to compare sequences.

## ☒ **Compare Triggers**

Check this option if you want to compare triggers.

## ☒ **Compare Rules**

Check this option if you want to compare rules.

## **Execution Options**

### ☒ **SQL for objects to be created**

Check this option to include all related SQL statements if new objects will be created in the target database and schema.

### ☒ **SQL for objects to be changed**

Check this option to include all related SQL statements if objects will be changed in the target database and schema.

### ☒ **SQL for objects to be dropped**

Check this option to include all related SQL statements if objects will be dropped from the target database and schema.

### ☒ **Drop with CASCADE**

Check this option if you want to cascade to drop the dependent objects.

### ☒ **Compare after execution**

Compares tables after the synchronization is executed.

### ☒ **Continue on error**

Ignores errors that are encountered during the synchronization process.

### ☒ **Create inheriting parent**

Creates tables of inheriting parents during the synchronization process.

## General Settings for SQL Server Structure Synchronization

The following instruction guides you through the process of setting up a structure synchronization. Customize options according to your needs.

### Source

Defines connection, database and schema for the source.

### Target

Defines connection, database and schema for the target.

### Compare Options

#### ☒ **Compare Tables**

Check this option if you want to compare tables between the source and target databases. Select/unselect the six options below:

#### ☒ **Compare Primary Keys**

Check this option if you want to compare table primary keys.

#### ☒ **Compare Foreign Keys**

Check this option if you want to compare table foreign keys.

#### ☒ **Compare Uniques**

Check this option if you want to compare uniques.

#### ☒ **Compare Checks**

Check this option if you want to compare checks.

#### ☒ **Compare Collation**

Check this option if you want to compare collation of the tables.

#### ☒ **Compare Identity Last Value**

Check this option if you want to compare table identity last values.

#### ☒ **Compare Views**

Check this option if you want to compare views.

#### ☒ **Compare Functions**

Check this option if you want to compare functions.

## ☒ **Compare Indexes**

Check this option if you want to compare indexes.

## ☒ **Compare Triggers**

Check this option if you want to compare triggers.

## ☒ **Compare Storage**

Check this option if you want to compare storage.

## **Execution Options**

### ☒ **SQL for objects to be created**

Check this option to include all related SQL statements if new objects will be created in the target database and schema.

### ☒ **SQL for objects to be changed**

Check this option to include all related SQL statements if objects will be changed in the target database and schema.

### ☒ **SQL for objects to be dropped**

Check this option to include all related SQL statements if objects will be dropped from the target database and schema.

### ☒ **Compare after execution**

Compares tables after the synchronization is executed.

### ☒ **Continue on error**

Ignores errors that are encountered during the synchronization process.

## Structure Synchronization Result

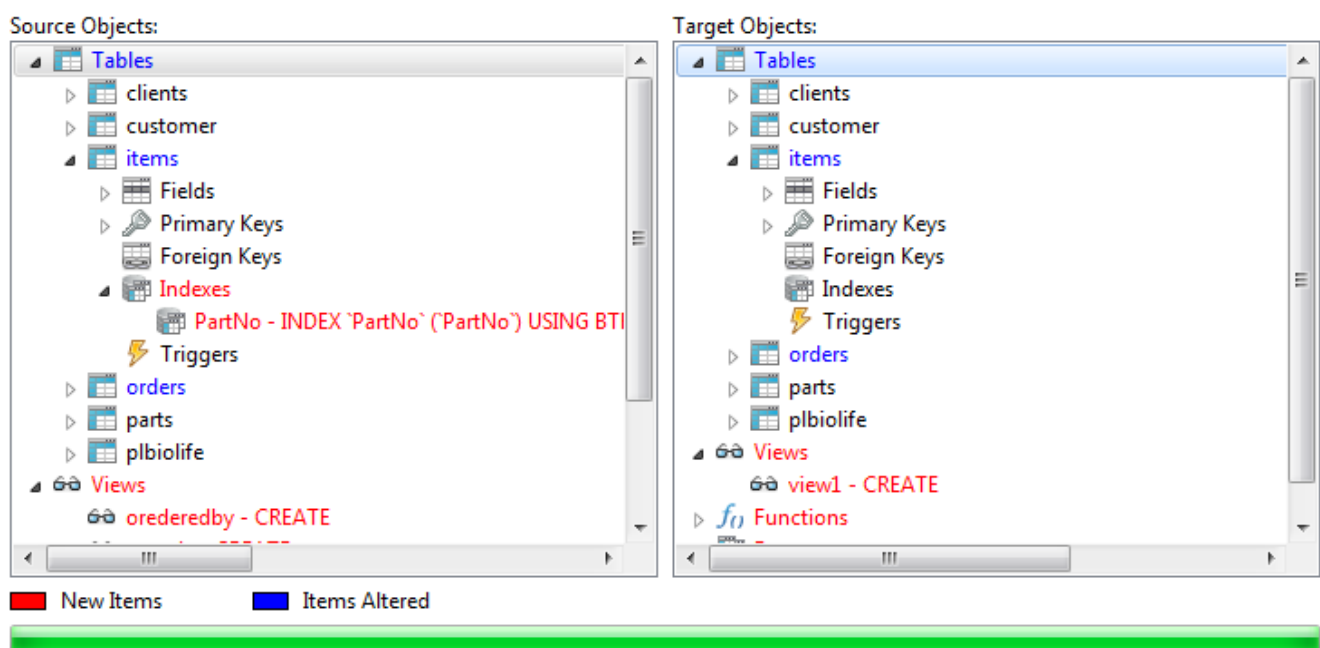
### Source Objects/Target Objects

The tree view shows the differentiation between the source database/schema and target database/schema after the computation of the structure synchronization, providing with the detailed SQL statements shown in the **Queries for Modification** list.

The red item represents the non-existence for the other database/schema.

The blue item represents the existence for the other database/schema, but different definition detected.

You are allowed to edit the object structure manually, just simply right-click the object in the tree view and click **Edit** to open the relevant designer.

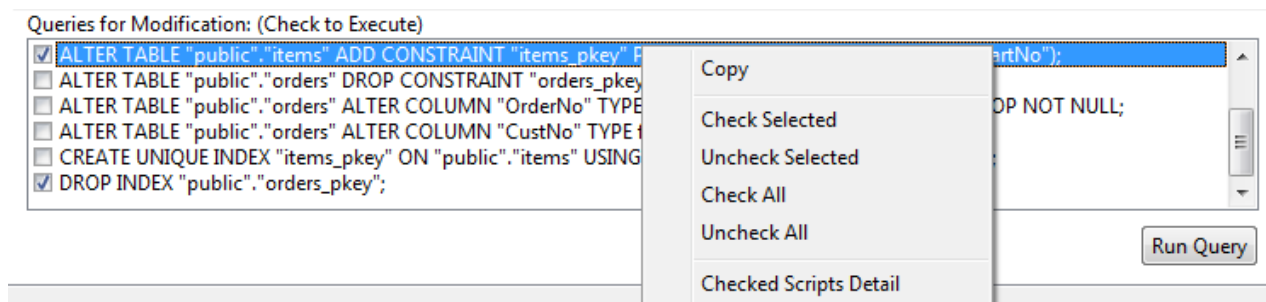


### Queries for Modification

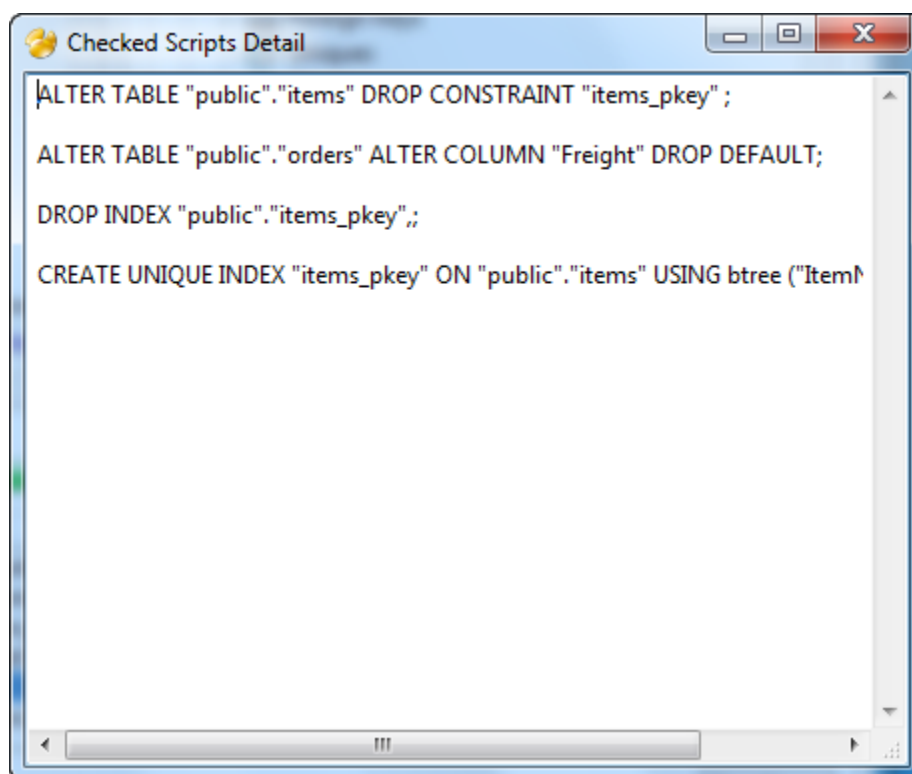
All the scripts are applied to the target database/schema and they are being unchecked in the **Queries for Modification** list by default. Just simply select the scripts you want to execute.

You can highlight multiple lines of scripts, and then right-click to show the pop-up menu. Choose **Copy** can copy the selected queries to preferred editor. Choose **Check Selected**, **Uncheck Selected**, **Check All** or **Uncheck All** so as to perform selection/unselection of scripts at one go.





To view the full SQL statements you selected, right-click the **Queries for Modification** list and select **Checked Scripts Detail**.



Press **Run Query** to execute the selected query.

## Structure Synchronization Message Log

The **Message Log** tab allows you to view the running process indicating success or failure.


Example:

```
[Msg] Starting Synchronization
[Msg] Source Server : Localhost
[Msg] Source Database : report_sample
[Msg] Target Server : Localhost
[Msg] Target Database : report_backup
[Msg] Executing - ALTER Table 'clients' MODIFY COLUMN 'RecordID' int(10) NOT NULL
auto_increment;
[Msg] Completed
[Msg] Synchronization Completed
```

## Backup/Restore (Available only in Full version & only for MySQL, PostgreSQL and SQLite)



A secure and reliable server is closely related to performing regular backups, as failures will probably occur sometimes - caused by attacks, hardware failure, human error, power outages, etc.


Navicat allows you to backup/restore all tables and records, views and functions for your database. Backup can be invoked from the command line, which makes it possible to schedule backups between databases.

Just simply click  to open an object pane for **Backup**. A right-click displays the popup menu or using the backup object pane toolbar, allowing you to create new, restore, extract and delete the backup.

### Create Backup

To create a new backup

- Select anywhere on the object pane.
- Click the  **New Backup** from the object pane toolbar.  
or
- Right-click and select  **New Backup** from the popup menu.
- Edit backup properties on the appropriate tabs.
- Click **Start**.

**Hint:** To create new backup you can also right-click the Backups node of the navigation pane and select the  **New Backup** from the popup menu.

### Edit Backup

To change the name of the backup

- Select the backup for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.



## Compress or Decompress Backup

To compress or decompress the backup

- Select the backup for compressing or decompressing in the navigation pane/object pane.
- Right-click and select the **Compress Backup** or **Decompress Backup** from the popup menu.



## Delete Backup



To delete a backup

- Select the backup for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Backup** from the popup menu.  
or
- Click the  **Delete Backup** from the object pane toolbar.
- Confirm deleting in the dialog window.




## Restore Backup

To restore a backup to an existing database

- Open the database and select the existing backup.
- Click the  **Restore Backup** from the object pane toolbar or right-click and select  **Restore Backup** from the popup menu.
- Edit restore options on the appropriate tabs.
- Click **Start**.



**Hint:** To restore the backup you can also click the  **Restore Backup** from the object pane toolbar or right-click the Backup node of the navigation pane and select the  **Restore Backup** from the popup menu.

To restore a backup to a new database

- Create a new database (MySQL, PostgreSQL or SQLite) and click  to open an object pane for Backup.
- Click the  **Restore Backup** from the object pane toolbar or right-click anywhere on the object pane and select  **Restore Backup** from the popup menu.
- Browse the backup file.
- Edit restore options on the appropriate tabs.
- Click **Start**.

## Extract SQL

To extract backup to sql file

- Select the backup for extracting in the navigation pane/object pane.
- Right-click and select the  **Extract SQL** from the popup menu.  
or
- Click the  **Extract SQL** from the object pane toolbar.
- Edit extract SQL options on the appropriate tabs.
- Click **Start**.

## Achieve Backup Information

To achieve a backup information (Name, Group Name and File Size, etc)

- Select the backup in the navigation pane/object pane.
- Right-click the selected backup and choose **Object Information** from the popup menu to view the Object Information.  
or
- Choose View -> Object Information in the main menu.

## Backup

**Backup** is the basic Navicat tool for performing regular backups.

- [General Settings for Backup](#)
- [Object Selection for Backup](#)
- [Advanced Settings for Backup](#)
- [Message Log](#)

**Hint:** Backup files are stored under Settings Save Path.

To run a backup from the command line

- Create and save the backup profile.
- Start Navicat from command line, type the command (see Command for details)

## General Settings for Backup

### Backup File Option

#### **Comment**

Allows you to enter the comment for the backup.

## **Object Selection for Backup**

You are allowed to choose your preferable database objects, i.e. tables, views, functions, sequences, events, indexes and triggers you wish to backup.



## Advanced Settings for Backup

### ☒ **Compressed**

Check this option if you want to produce smaller backup size.

**Hint:** compressed (.psc), uncompressed (.psb).

### ☒ **Lock All Tables (Available only for MySQL and PostgreSQL)**

Lock all objects while backup is being processed.

### ☒ **Use Single Transaction (InnoDB only) (Available only for MySQL)**

If a table uses InnoDB storage engine, with this option is on, Navicat uses transaction before the backup process starts.

### ☒ **Use specify file name**

Define your file name for backup. Otherwise, your backup file will be named as **"2007-05-10 17:38:20"** for example.

## Backup Message Log

The **Message Log** tab allows you to view the running process indicating success or failure.

Example:

```
[Msg] [Bak] Starting backup...
[Msg] [Bak] Writing file header...
[Msg] [Bak] Writing structure header...
[Msg] [Bak] Writing schema dummy header...
[Msg] [Bak] Writing table clients header...
[Msg] [Bak] Writing table customer header...
[Msg] [Bak] Writing table items header...
[Msg] [Bak] Writing table orders header...
[Msg] [Bak] Writing table parts header...
[Msg] [Bak] Writing table plbiolife header...
[Msg] [Bak] Writing view view_mysql1 header...
[Msg] [Bak] Prepare writing data...
[Msg] [Bak] Writing data...
[Msg] [Bak] Writing table clients data...
[Msg] [Bak] Writing table customer data...
[Msg] [Bak] Writing table items data...
[Msg] [Bak] Writing table orders data...
[Msg] [Bak] Writing table parts data...
[Msg] [Bak] Writing table plbiolife data...
[Msg] [Bak] Compressing backup file...
[Msg] [Bak] Finished - Successfully
```

## Restore

Navicat provides a useful tool for restoring your backup while hardware failure occurs.

- [General Settings for Restore](#)
- [Object Selection for Restore](#)
- [Advanced Settings for Restore](#)
- [Message Log](#)

**Note:** You must have Create, Drop and Insert Privileges (MySQL or PostgreSQL) to run the restore.

**Hint:** Restore function will firstly drop the selected objects of the database, then recreate the new objects according to your backup. Finally, inserting the data.

## General Settings for Restore

### Backup file information

#### Comment

Allows you to enter the comment for the restore.

## **Object Selection for Restore**

You are allowed to choose your preferable database objects, i.e. tables, views, functions, sequences, events, indexes and triggers you wish to restore.

## Advanced Settings for Restore

### Server Options

☒ **Use Transaction**

Rollbacks all data when error occurs.

☒ **Continue on error**

Ignores errors that are encountered during the restore process.

☒ **Lock tables for write (Available only for MySQL and PostgreSQL)**

Locks the tables to prevent user to modify tables during the restore process.

☒ **Use extended insert statements (Available only for MySQL)**

Inserts records using extended insert syntax.

Example:

```
INSERT INTO `users` VALUES ('1', 'Peter McKindy', '23'), ('2', 'Johnson Ryne', '56'), ('0', 'Katherine', '23');
```

☒ **Run multiple queries in each execution (Available only for PostgreSQL)**

Checks this option if you want to run multiple queries in each execution, which will make the restore process faster.

### Object Options

☒ **Create tables**

Creates tables during the restore process with this option is on.

☒ **Create records**

Restores records during the restore process with this option is on, otherwise, only table structures will be restored.

☒ **Create indexes (Available only for PostgreSQL)**

Creates indexes for the restored table with this option is on.

☒ **Create triggers (Available only for MySQL and PostgreSQL)**

Creates triggers for the restored table with this option is on.

☒ **Overwrite existing tables**

Overwrites if tables already exist in the database.

☒ **Overwrite existing views**

Overwrites if views already exist in the database.

☒ **Overwrite existing functions (Available only for MySQL and PostgreSQL)**

Overwrites if functions already exist in the database.

☒ **Overwrite existing events (Available only for MySQL)**

Overwrites if events already exist in the database.

☒ **Overwrite existing sequences (Available only for PostgreSQL)**

Overwrites if sequences already exist in the database.

☒ **Overwrite existing indexes(Available only for SQLite)**

Overwrites if indexes already exist in the database.

☒ **Overwrite existing triggers (Available only for SQLite)**

Overwrites if triggers already exist in the database.

☒ **Insert Auto Increment Values (Available only for SQLite)**

Inserts auto increment values in the database.

## Restore Message Log

The **Message Log** tab allows you to view the running process indicating success or failure.

Example:

```
[Msg] Decompressing...
[Msg] Table Created: clients
[Msg] Table Created: customer
[Msg] Table Created: items
[Msg] Table Created: orders
[Msg] Table Created: parts
[Msg] Table Created: plbiolife
[Msg] Importing Data...
[Msg] Table Restored: clients
[Msg] Table Restored: customer
[Msg] Table Restored: items
[Msg] Table Restored: orders
[Msg] Table Restored: parts
[Msg] Table Restored: plbiolife
[Msg] Finished successfully
```



## Extract SQL

You are allowed to extract your backup into a SQL file. See Restore for details.


Example:

```
[Msg] Decompressing...
[Msg] Table DDL Extracted: clients
[Msg] Table DDL Extracted: customer
[Msg] Table DDL Extracted: items
[Msg] Table DDL Extracted: orders
[Msg] Table DDL Extracted: parts
[Msg] Table DDL Extracted: plbiolife
[Msg] Table Data Extracted: clients
[Msg] Table Data Extracted: customer
[Msg] Table Data Extracted: items
[Msg] Table Data Extracted: orders
[Msg] Table Data Extracted: parts
[Msg] Table Data Extracted: plbiolife
[Msg] Finished - Successfully
```





## Batch Job/Schedule (Available only in Full Version)

Navicat allows you to create a batch job for setting schedule to execute at one or more regular intervals, beginning and ending at a specific date and time using **Windows Task Scheduler**. Batch job can be created for Query, Report printing, Backup, Data Transfer, Data Synchronization, Import and Export from MySQL, Oracle, PostgreSQL, SQLite and SQL Server. You can define a list of actions to be performed within one batch job, either run it manually or at the specified time/periodically.




Just simply click  to open an object pane for **Schedule**. A right-click displays the popup menu or using the schedule object pane toolbar, allowing you to create new, edit, open and delete the selected batch job/schedule.

### Create Batch Job

To create a new batch job



- Select anywhere on the object pane.
- Click the  **New Batch Job** from the object pane toolbar.  
or
- Right-click and select  **New Batch Job** from the popup menu.
- Edit batch job properties on the appropriate tabs.

To create a new batch job with modification as one of the existing batch jobs

- Select the batch job for modifying in object pane.
- Right-click and select the  **Design Batch Job** from the popup menu.  
or
- Click the  **Design Batch Job** from the object pane toolbar.
- Modify batch job properties on the appropriate tabs.
- Click  **Save As**.

## Edit Batch Job

To edit the existing batch job


- Select the batch job for editing in the object pane.
- Right-click and select the  **Design Batch Job** from the popup menu.  
or
- Click the  **Design Batch Job** from the object pane toolbar.
- Edit batch job properties on the appropriate tabs.

To change the name of the batch job

- Select the batch job for editing in the object pane.
- Right-click and select the **Rename** from the popup menu.

## Run Batch Job

To run a batch job



- Create a new batch job/open the existing one.
- Click  **Start**.

To run a saved batch job from the command line

- Create and save the batch job.
- Start Navicat from command line, type the command (see Command for details)

## Delete Batch Job

To delete a batch job

- Select the batch job for deleting in the object pane.
- Right-click and select the  **Delete Batch Job** from the popup menu.  
or
- Click the  **Delete Batch Job** from the object pane toolbar.
- Confirm deleting in the dialog window.



## Convert Batch Job

To convert a batch job

- Right-click and select the **Batch Job Converter** from the popup menu in the object pane.
- Select the batch jobs.
- Set the convert options.
- Click **Start**.

## Set Schedule



To set schedule to the batch job

- Create and save the batch job/open the existing one.
- Select the batch job in the object pane.
- Right-click and select the  **Set Task Schedule** from the popup menu.  
or
- Click the  **Set Task Schedule** from the object pane toolbar.
- Set your schedule using Windows Scheduler.

**Hint:** LogCmd.txt stores all the operations executed, indicating success or failure during the schedule.

## Delete Schedule

To delete a schedule

- Select the scheduled batch job in the object pane.
- Right-click and select the  **Delete Task Schedule** from the popup menu.  
or
- Click the  **Delete Task Schedule** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Batch Job Information

To achieve a batch job information (Name, Group name, File and Create Time, etc)

- Select the batch job in the object pane.
- Right-click the selected batch job and choose **Object Information** from the popup menu to view the Object Information.  
or
- Choose View -> Object Information in the main menu.

### General Settings for Batch Job/Schedule

The following instruction guides you through the process of setting up a batch job/schedule. Customize options according to your needs.

Move the objects from the **Available Jobs** list to the **Selected Jobs** list using **Select../Unselect..** buttons, by double-clicking or dragging them. To delete the objects from the selected jobs list, remove them in the same way. You are allowed to run profiles from different servers in a single batch job/schedule.

You are allowed to group jobs into a procedure, running them in sequence, each starting the next. To rearrange the sequence of the jobs, select the **Move Up/Move Down** buttons.

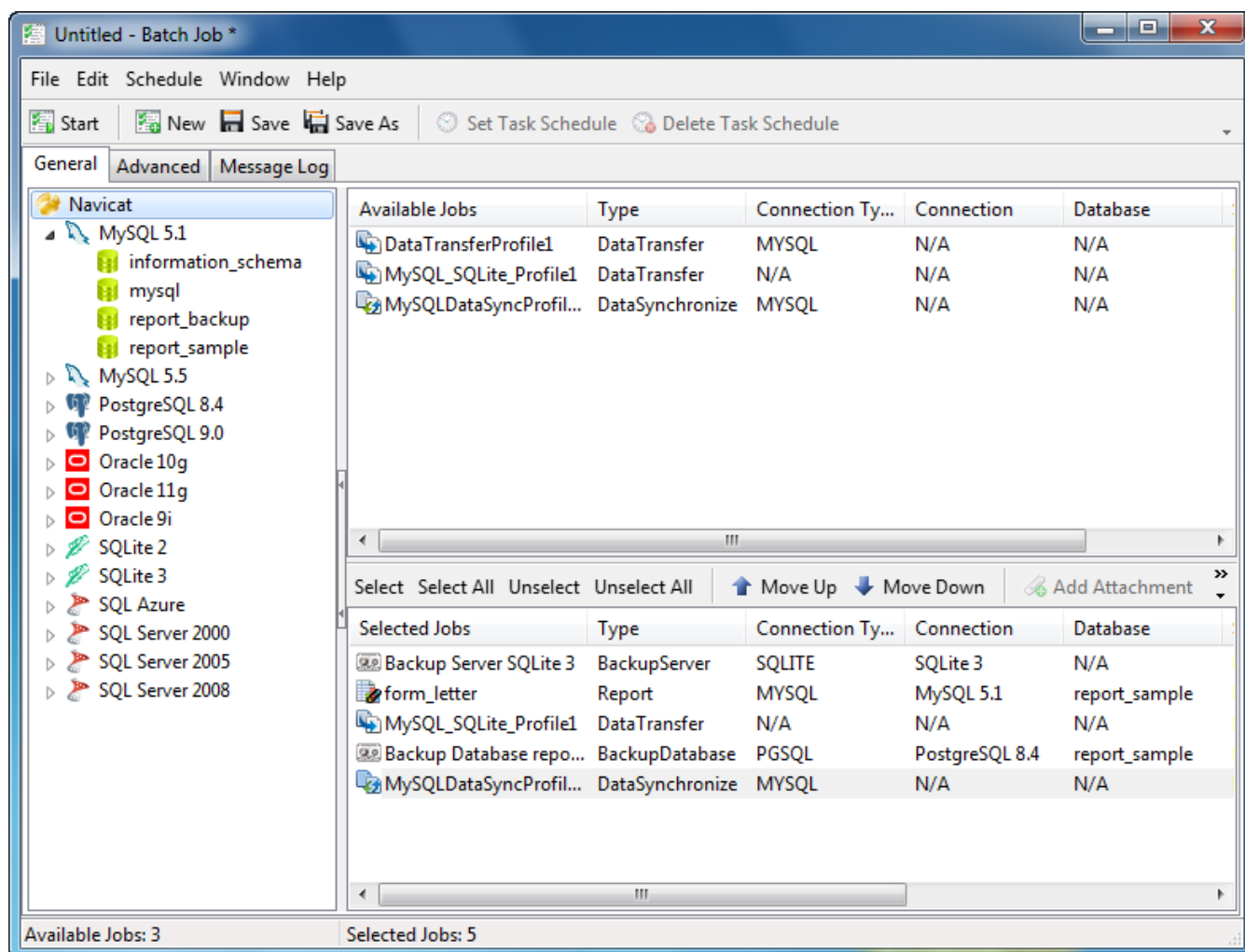
Exported file or printed report can be added to the batch job as mail attachment. Simply select the job in Selected Jobs and click **Add Attachment** or **Remove Attachment** to add or remove the mail attachment. (To set up the mail sending, see Advanced Settings for Batch Job/Schedule.)

You are allowed to backup server/multiple servers, just simply select the servers from the left panel and move the **Backup Server xxx..** from the **Available Jobs** list to the **Selected Jobs** list. (To backup your connection settings, see registry.)

To set schedule for running Data Transfer or Data Synchronization profile, choose **Navicat** at the top on the left panel.

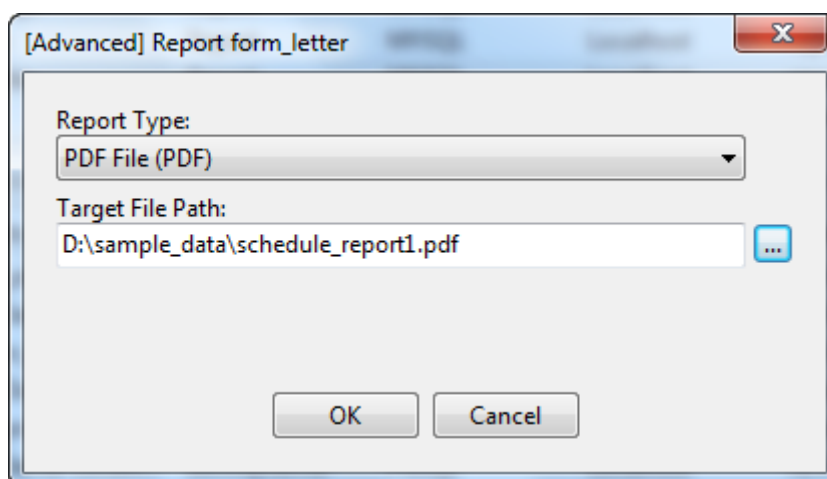
**Note:** Please save the batch job before setting schedule.

<b>Note:</b> Passwords must be saved in Connection Properties (MySQL, Oracle, PostgreSQL or SQL Server) and Windows Scheduler before running your schedule.	Connection Properties	Windows Scheduler
	<div>User Name:<div>root</div></div> <div>Password:<div>.....</div></div> <div><input checked="" type="checkbox"/> Save Password</div>	<div>Start in:<div>"D:\Navicat Win Version 9\"</div></div> <div>Comments:<div></div></div> <div>Run as:<div>Siuha-win7\Test</div><div>Set password...</div></div>



## Setting Report Printing

Navicat supports to make schedule for printing your report to physical printer or in multiple format, e.g. Excel, HTML, PDF and more. Just simply move the saved report(s) from the **Available Jobs** list to the **Selected Jobs** list. Customize options in the **Report** dialog window according to your needs.





## Advanced Settings for Batch Job/Schedule

Navicat allows you to generate and send personalized e-mails with results returned from a schedule. The resultset(s) can be emailed to multiple recipients.

### ☒ **Send Email**

#### **From**

Specifies the e-mail address that people should use when sending e-mail to you at this account. For example, someone@navicat.com.

#### **To, CC**

Specifies the e-mail address of each recipient, separating them with a comma or a semicolon (;).

#### **Host (SMTP Server)**

Enters your Simple Mail Transfer Protocol (SMTP) server for outgoing messages.

#### **Port**

Enters the port number you connect to on your outgoing e-mail (SMTP) server. Default value is **25**.

### ☒ **Use Authentication**

Check this option and enter required user name and password if your SMTP server requires authorization to send email.

#### **Secure Connection**

Specifies the connection to use TLS, SSL secure connection or not.

#### **Send test mail**

Navicat will send you a test mail indicating success or failure.

## Batch Job/Schedule Message Log

The **Message Log** tab allows you to view the running process indicating success or failure.

**Note:** LogCmd.txt stores all the operations executed during the schedule.

Example:

```
[Msg] Batch job started
[Msg] [Bak] Backup Localhost->report_sample
[Msg] [Bak] Starting backup...
[Msg] [Bak] Writing header...
[Msg] [Bak] Writing Data...
[Msg] [Bak] Compressing Backup File...
[Msg] [Rep] Report Localhost->report_sample->invoice
[Msg] [Rep] Finished - Successfully
[Msg] [Que] Query Localhost->report_sample->query_2
[Msg] [Que] Finished - Successfully
[Msg] Finished - Successfully
```

## Batch Job Converter (Available only in Navicat Premium)

Navicat Premium allows you to convert saved batch jobs from Navicat for MySQL, Navicat for Oracle, Navicat for PostgreSQL, Navicat for SQLite and Navicat for SQL Server to it.

A right-click in **Schedule** object pane and select **Batch Job Converter...** from the popup menu to open the batch job converter window.

- [Selecting batch jobs](#)
- [Setting convert options](#)
- [Starting convert](#)

## Selecting Batch Jobs

Select batch jobs to convert.

### Select All

You can select all batch jobs by simply selecting **Select All** button for quick mapping.

In Vista or above, if you select batch jobs from either one Navicat, you can just select **Select All from Navicat for MySQL** or **Select All from Navicat for PostgreSQL** or **Select All from Navicat for Oracle** or **Select All from Navicat for SQLite** or **Select All from Navicat for SQL Server** from **Select All** button for quick mapping.

### Unselect All

You can unselect all selected batch jobs easily.

## Setting Convert Options

### Options

☒ **Delete original batch jobs**

Check this option if you want to delete the original batch jobs in Navicat. If the original batch job is deleted, the scheduled batch job will not work until it is set again in Navicat Premium or the original application.

☒ **Overwrite existing batch jobs**

Check this option if you want to overwrite the existing batch jobs in Navicat Premium.

☒ **Append when batch job exists**

Check this option and enter the name of existing batch job if you want to append the details to the existing batch jobs in Navicat Premium.

## Starting Convert

You can view the running process indicating success or failure.

Example:

```
----- Batch job conversion starts -----  
[Msg] Converting batch jobs: "postgresql_schedule1"...  
[Msg] "postgresql_schedule1" created  
----- Batch job conversion done -----  
Batch job created: 1  
Batch job overwritten: 0  
Batch job renamed: 0  
Batch job conversion failed: 0  
Old batch job deleted: 0
```

## Console

Navicat provides interactive text-based screen for user query input and result output from MySQL, Oracle, PostgreSQL, SQLite and SQL Server.

- [MySQL Console](#)
- [Oracle Console](#)
- [PostgreSQL Console](#)
- [SQLite Console](#)
- [SQL Server Console](#)


## MySQL Console


**MySQL Console** allows you to use a command-line interface. In other words, it provides interactive text-based screen for you to query input and result output from database.

Just simply open the console and use the console toolbar, allowing you to run, save and load your SQL statements.

### Open Console

To open a console window

- Open the connection and select Tools ->  **Console...** from the main menu or just simply press **F6**.
- Edit your SQL statement in the console.

**Hint:** To create new console you can also right-click the Database node of the navigation pane and select the  **Console...** from the popup menu.



To open a console window with loading from a text file

- Open the connection and select Tools ->  **Console...** from the main menu or just simply press **F6**.
- Click  **Load**.

**Hint:** You are allowed to open multiple console windows which each represents different connection.

### Save Statements

To save the SQL statement into text file

- Open the connection and select Tools ->  **Console...** from the main menu or just simply press **F6**.
- Edit your SQL statement in the console.
- Click  **Save**.



## Stop Executing

To stop running the SQL statement

- Click  **Stop**.

## Erase Content

To clear all SQL statements in console window

- Click  **Clear**.

## Exit Console

To exit a console window

- Click the cross button at the main bar.  
or
- Type **quit** in the console window and press Enter.

## Example of Using MySQL Console

### Basic MySQL query statements

```
mysql> show databases;
```

```
+-----+
| Database          |
+-----+
| information_schema |
| mysql              |
| report_backup      |
| report_sample      |
+-----+
```

4 rows in set

```
mysql> use report_sample;
```

Database changed

```
mysql> show tables;
```

```
+-----+
| Tables_in_report_sample |
+-----+
| clients                  |
| customer                 |
| items                    |
| orders                   |
| parts                    |
| plbiolife                |
+-----+
```

6 rows in set

```
mysql> select Description, Cost from parts where PartNo = 2619;
```

```
+-----+-----+
| Description          | Cost |
+-----+-----+
| Navigation Compass  | 9.177 |
+-----+-----+
```

1 row in set

```
mysql>
```

## Oracle Console

**Oracle Console (SQL\*Plus)** allows you to use a command-line interface. In other words, it provides interactive text-based screen for you to query input and result output from database.

**Note:** You have to have SQL\*Plus executable in order to get this works. If it cannot be found under the default path, you are prompted to locate the executable in the pop up.

## Downloading

**SQL\*Plus** is included in packages **Oracle Client** / **Oracle Instant Client**. You can download it through -

### Oracle Client

<http://www.oracle.com/technology/software/products/database/index.html>


### Oracle Instant Client

<http://www.oracle.com/technology/software/tech/oci/instantclient/index.html>

## Open Console

To open console window

- Open the connection and select **Tools** ->  **Console...** from the main menu or just simply press **F6**.
- Edit your SQL statement in the console command prompt.

**Hint:** To open a new console window you can also right-click the Connection/Schema node of the navigation pane and select the  **Console...** from the popup menu.

**Hint:** You are allowed to open multiple console windows which represents different connection each.

## Exit Console

To exit console

- Click the cross button at the main bar.  
or
- Type **exit** or **quit** in the console window and press Enter.

See also:  
OCI Option

## Example of Using Oracle Console

### Basic Oracle query statements

```
SQL> select DEPARTMENT_ID, DEPARTMENT_NAME from DEPARTMENTS where  
LOCATION_ID = 1700;
```

```
DEPARTMENT_ID DEPARTMENT_NAME
```

```
-----
```

```
10 Administration  
90 Executive  
100 Finance  
110 Accounting  
120 Treasury  
270 Payroll
```

```
6 rows selected.
```

```
SQL>
```


## PostgreSQL Console


**PostgreSQL Console** allows you to use a command-line interface. In other words, it provides interactive text-based screen for you to query input and result output from database.

Just simply open the console and use the console toolbar, allowing you to run, save and load your SQL statements.

### Open Console

To open a console window

- Open the connection and select **Tools** ->  **Console...** from the main menu or just simply press **F6**.
- Edit your SQL statement in the console.

**Hint:** To create new console you can also right-click the Database/Schema node of the navigation pane and select the  **Console...** from the popup menu.



To open a console window with loading from a text file

- Open the connection and select **Tools** ->  **Console...** from the main menu or just simply press **F6**.
- Click  **Load**.

**Hint:** You are allowed to open multiple console windows which represents different connection.

### Save Statements

To save the SQL statements into text file

- Open the connection and select **Tools** ->  **Console...** from the main menu or just simply press **F6**.
- Edit your SQL statements in the console.
- Click  **Save**.
- Browse a destination to save the text file.

## Stop Executing

To stop running the SQL statement

- Click  **Stop**.

## Erase Content

To clear all SQL statements in console window

- Click  **Clear**.

## Exit Console

To exit a console window

- Click the cross button at the main bar.  
or
- Type **quit** in the console window and press Enter.

## Example of Using PostgreSQL Console

### Basic PostgreSQL query statements

```
report_sample=# select datname from pg_database;
```

```
+-----+
| datname      |
+-----+
| template1    |
| template0    |
| postgres     |
| report_sample|
| report_backup|
+-----+
```

5 rows in set

```
report_sample=# select tablename from pg_tables;
```

```
+-----+
| tablename          |
+-----+
| customer           |
| clients            |
| items             |
| orders            |
| parts             |
| plbiolife          |
+-----+
```

6 rows in set

```
report_sample=# select "Description", "Cost" from public.parts where "PartNo" = 2619;
```

```
+-----+-----+
| Description      | Cost |
+-----+-----+
| Navigation Compass | 9.177 |
+-----+-----+
```

1 row in set

```
report_sample=#
```




## SQLite Console


**SQLite Console** allows you to use a command-line interface. In other words, it provides interactive text-based screen for you to query input and result output from database.

Just simply open the console and use the console toolbar, allowing you to run, save and load your SQL statements.

### Open Console

To open a console window

- Open the connection and select Tools ->  **Console...** from the main menu or just simply press **F6**.
- Edit your SQL statement in the console.

**Hint:** To create new console you can also right-click the Database node of the navigation pane and select the  **Console...** from the popup menu.



To open a console window with loading from a text file

- Open the connection and select Tools ->  **Console...** from the main menu or just simply press **F6**.
- Click  **Load**.

**Hint:** You are allowed to open multiple console windows which each represents different connection.

### Save Statements

To save the SQL statement into text file

- Open the connection and select Tools ->  **Console...** from the main menu or just simply press **F6**.
- Edit your SQL statement in the console.
- Click  **Save**.

## Erase Content

To clear all SQL statements in console window

- Click  **Clear**.

## Exit Console

To exit a console window

- Click the cross button at the main bar.  
or
- Type **quit** in the console window and press Enter.

## Example of Using SQLite Console

### Basic SQLite query statements

```
sqlite> pragma database_list;
```

```
+-----+-----+-----+
| seq | name | file |
+-----+-----+-----+
| 0   | main | D:\sqlite-3_6_22\sqlite_table.db |
+-----+-----+-----+
```

1 rows in set

```
sqlite> select Description, Cost from parts where PartNo = 2619;
```

```
+-----+-----+
| Description      | Cost |
+-----+-----+
| Navigation Compass | 9.177 |
+-----+-----+
```

1 rows in set

```
sqlite>
```


## SQL Server Console


**SQL Server Console** allows you to use a command-line interface. In other words, it provides interactive text-based screen for you to query input and result output from database.

Just simply open the console and use the console toolbar, allowing you to run, save and load your SQL statements.

### Open Console

To open a console window

- Open the connection and select Tools ->  **Console...** from the main menu or just simply press **F6**.
- Edit your SQL statement in the console.

**Hint:** To create new console you can also right-click the Database node of the navigation pane and select the  **Console...** from the popup menu.



To open a console window with loading from a text file

- Open the connection and select Tools ->  **Console...** from the main menu or just simply press **F6**.
- Click  **Load**.

**Hint:** You are allowed to open multiple console windows which each represents different connection.

### Save Statements

To save the SQL statement into text file

- Open the connection and select Tools ->  **Console...** from the main menu or just simply press **F6**.
- Edit your SQL statement in the console.
- Click  **Save**.

## Stop Executing

To stop running the SQL statement

- Click  **Stop**.

## Erase Content

To clear all SQL statements in console window

- Click  **Clear**.

## Exit Console

To exit a console window

- Click the cross button at the main bar.  
or
- Type **quit** in the console window and press Enter.

## Example of Using SQL Server Console

### Basic SQL Server query statements

```
1> use AdventureWorks;
2> select DepartmentID, Name from HumanResources.Department where
   GroupName = 'Executive General and Administration';
3> go
```

```
+-----+-----+
| DepartmentID | Name                               |
+-----+-----+
|          9 | Human Resources                   |
|         10 | Finance                           |
|         11 | Information Services              |
|         14 | Facilities and Maintenance        |
|         16 | Executive                         |
+-----+-----+
```

(5 rows affected)

```
1>
```

## Dump SQL File

Navicat allows you to backup your database/schema/table(s) using the **Dump SQL File** feature.

- Select the database/schema/table(s).
- Right-click and select **Dump SQL File...** from the popup menu.
- Save the sql file to your destination.

## Execute SQL File

Navicat allows you to restore your database/execute SQL file using the **Execute SQL File** feature.

- Select the database/schema.
- Right-click and select **Execute SQL File...** from the popup menu.




**Note:** You can drag a .sql file to the table pane or a database/schema in the connection tree. Navicat will popup the Execute SQL File.

Example:

[Msg] Finished - 35 queries executed successfully



## **Print Database/Schema/Table Structure (Available only in Full Version)**

Navicat allows you to view and print database, schema and table structure. Just simply right-click the database/schema/table(s) and select  **Print Database** or  **Print Schema** or  **Print Tables**.

## Log Files

Navicat provides number of log files to keep track on the actions have been performed in Navicat. Most of the log files are represented as text format, and they are located in the sub-directory called **logs**, e.g. C:\Users\Guest\Documents\Navicat\Premium\logs\. You are allowed to change the log files location under Options.

- **HttpDump.log**  
Stores information which response from your HTTP Server.
- **LogHistory.txt**  
Stores all SQL statements of all the operations executed over databases and database objects in Navicat.

**Note:** This log will be overwritten while Navicat being restarted.


**Hint:** Simply press Ctrl+H or click Tools > **History Log...** to open the LogHistory.txt file in the History Log Viewer.

- **LogImport.txt**  
Records detailed information on every error (indicating success or failure) that occurred during the import process.
- **LogExport.txt**  
Records detailed information on every error (indicating success or failure) that occurred during the export process.
- **LogSynchronize.txt**  
Records detailed information on every error (indicating success or failure) that occurred during the data synchronization process.

**Note:** This log will be overwritten on each synchronization.



- **LogCmd.txt**  
Stores all operations while running schedule.


## Report Management Tools (Available only in Navicat Premium and Enterprise Version)

Just simply click  to open an object pane for **Report**. A right-click displays the popup menu or using the object pane toolbar, allowing you to create new, view, edit and delete the selected report. You can save your reports for setting schedule.

### Create Report

To create a new report

- Select anywhere on the object pane.
- Click the  **New Report** from the object pane toolbar.  
or
- Right-click and select  **New Report** from the popup menu.
- Edit report on the appropriate tabs of the Report Builder.

**Hint:** To create new report you can also right-click the Reports node of the navigation pane and select the  **New Report** from the popup menu.

To create a new report with the same properties as one of the existing reports has (using popup menu)

**Apply to:** current database {same connection}

- Select the report(s) for copying in the navigation pane/object pane.
- Right-click and select the **Copy** from the popup menu.
- Select anywhere on the object pane.
- Right-click and select the **Paste** from the popup menu.
- The newly created report(s) will be named as "reportname - **Copy**".

To create a new report with the same properties as one of the existing reports has (using drag and drop method)




**Apply to:** current database {same connection}

- Select the report(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen report(s) to the target location.
- Select one of the following options:
  - Copy here
  - Move here
  - Cancel
- The newly created report(s) will be named as "reportname - **Copy**"

**Apply to:** different database {same connection}  
different database {different connection}



- Select the report(s) for copying in the navigation pane/object pane.
- Right-click and drag the chosen report(s) to the target location.
- Select one of the following options:
  - Copy here
  - Move here
  - Cancel

To create a new report with modification as one of the existing reports

- Select the report for modifying in the navigation pane/object pane.
- Right-click and select the  **Design Report** from the popup menu.  
or
- Click the  **Design Report** from the object pane toolbar.
- Modify report on the appropriate tabs of the Report Builder.
- Click  **Save As**.

## Edit Report

To edit the existing report



- Select the report for editing in the navigation pane/object pane.
- Right-click and select the  **Design Report** from the popup menu.  
or
- Click the  **Design Report** from the object pane toolbar.
- Modify report on the appropriate tabs of the Report Builder.

To change the name of the report



- Select the report for editing in the navigation pane/object pane.
- Right-click and select the **Rename** from the popup menu.

## View Report

To view a report

- Select the report for viewing in the navigation pane/object pane.
- Right-click and select the  **Open Report** from the popup menu or simply double-click the report.  
or
- Click the  **Open Report** from the object pane toolbar.

To view an archive report

- Right-click anywhere on the object pane and select the  **Open Report Archive** from the popup menu.  
or
- Click the  **Open Report Archive** from the object pane toolbar.
- Browse your archive file.

## Print Report (set schedule)

To print a report (using popup menu)

- Select the report for printing in the navigation pane/object pane.
- Right-click and select **Print Report** from the popup menu.

To print a report from the command line

- Create and save the report.
- In terminal, type the command (see Command for details)

## Print Report to File (set schedule)

To print report to file (using popup menu - PDF and HTML)

- Select the report for printing in the navigation pane/object pane.
- Right-click and select one of the options from the popup menu.
  - Print Report As Pdf
  - Print Report As HTML

To print report to file (setting under Report Tree windows - Text/Report Emulation Text, PDF, Excel, HTML/XHTML, RTF etc.)

**Hint:** **AllowPrintToFile** must be enabled in **Report Tree** windows to show a list of file formats that Report Builder supports.

- Select the report for printing in the navigation pane/object pane.
- Right-click and select the  **Design Report** from the popup menu.  
or
- Click the  **Design Report** from the object pane toolbar.
- In **Report Tree** windows, enable **AllowPrintToFile** under **Output - File** section.
- Choose the file type you wish to print in the Print Dialog.


## Print Report to Archive File

To print report to archive file (using popup menu)

- Select the report for printing in the navigation pane/object pane.
- Right-click and select **Print Report As Archive** from the popup menu.



To print report to archive file (setting under Report Tree windows)

**Hint:** **AllowPrintToArchive** must be enabled in **Report Tree** windows.

- Select the report for printing in the navigation pane/object pane.
- Right-click and select the  **Design Report** from the popup menu.  
or
- Click the  **Design Report** from the object pane toolbar.
- In **Report Tree** windows, enable **AllowPrintToArchive** under **Output - File** section.
- Choose **Archive File** in the Print Dialog.

## Delete Report

To delete a report

- Select the report for deleting in the navigation pane/object pane.
- Right-click and select the  **Delete Report** from the popup menu.  
or
- Click the  **Delete Report** from the object pane toolbar.
- Confirm deleting in the dialog window.

## Achieve Report Information

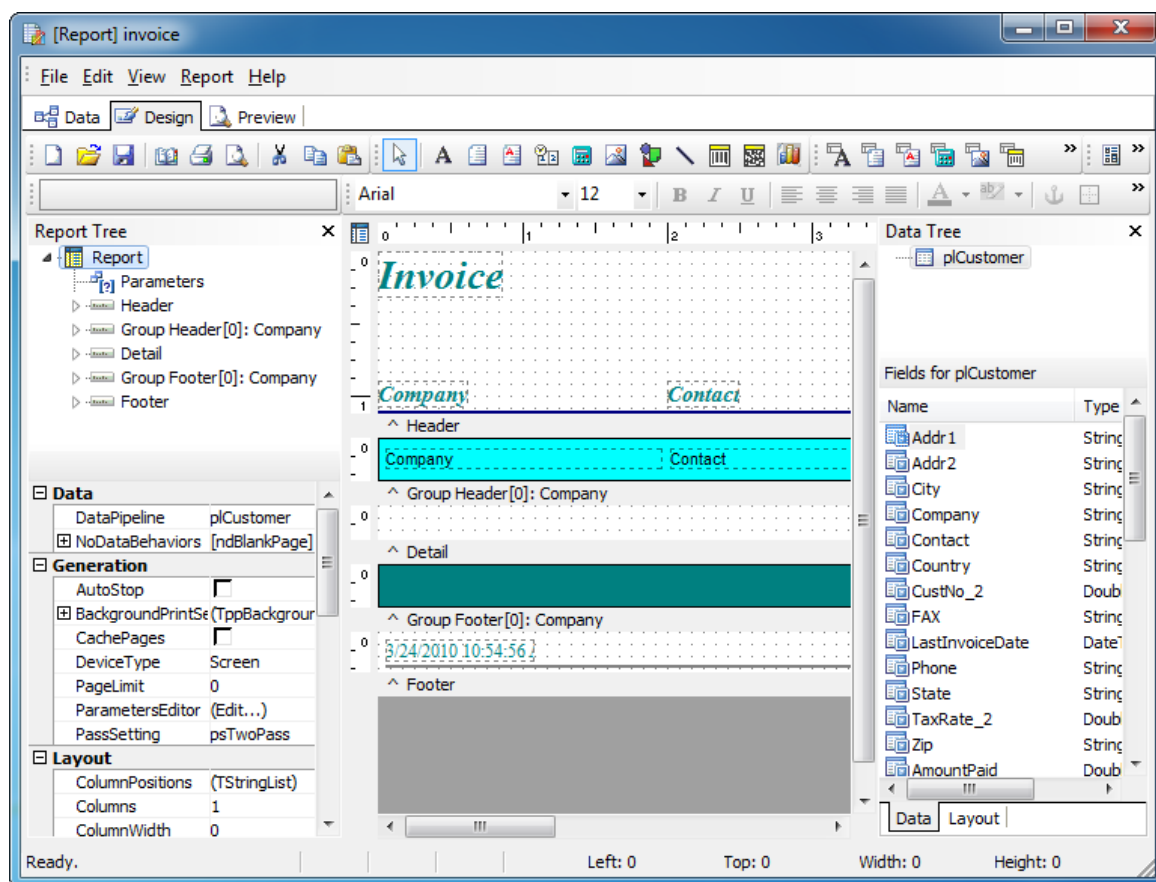
To achieve a report information

- Select the report in the object pane.
- Right-click the selected report and choose **Object Information** from the popup menu.  
or
- Choose View -> Object Information in the main menu.

## Report Designer

**Report Designer** is the basic Navicat tool for working with reports. It allows you to create new report and edit the existing report properties.

- [Report Data](#)
- [Report Design](#)
- [Report Preview](#)

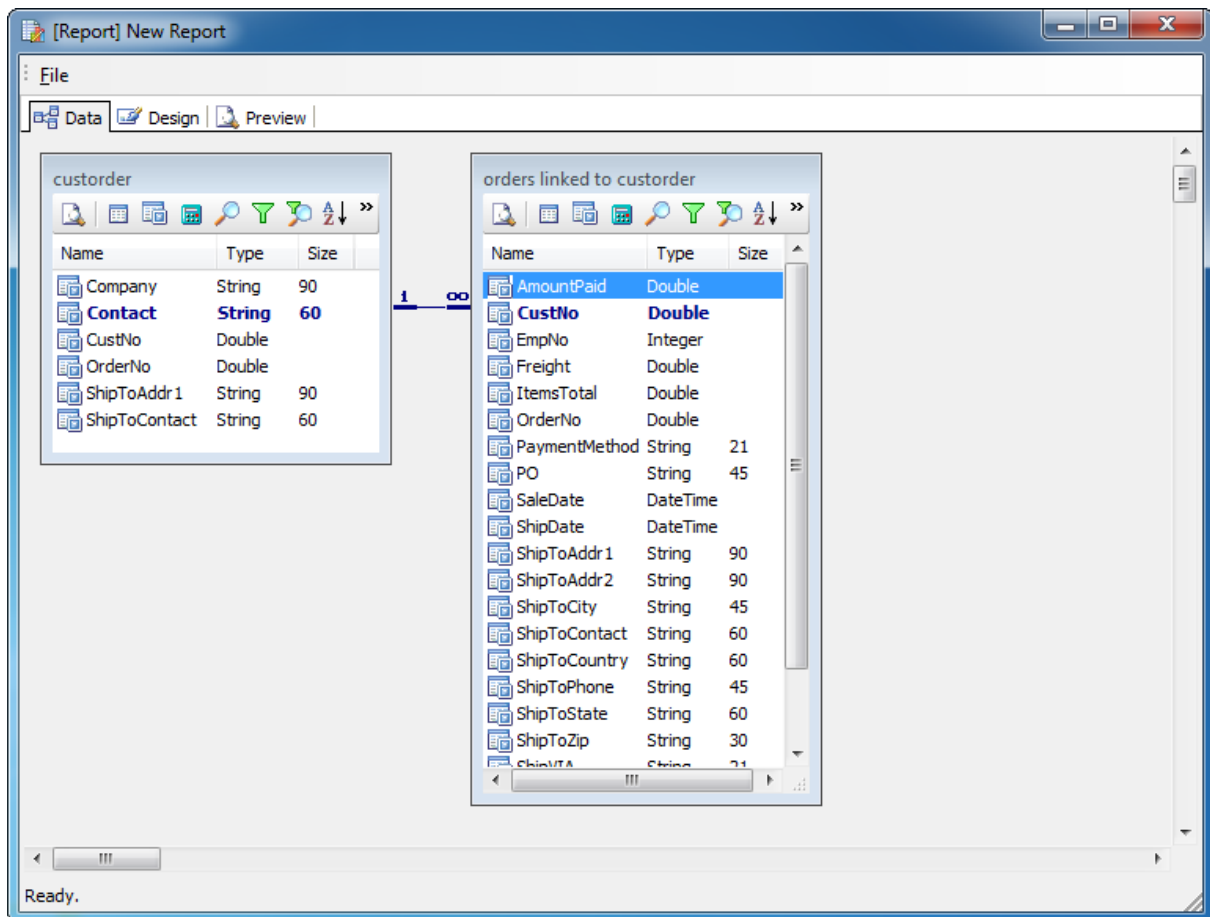




## Report Data

The **Data** tab allows us to select and manipulate the data needed for a report. These tasks are accomplished via two visual tools: the **Query Wizard** and the **Query Designer**. These tools greatly simplify the often-difficult task of data selection by giving us the ability to select data without requiring an in-depth knowledge of databases.

- [Query Wizard](#)
- [Query Designer](#)



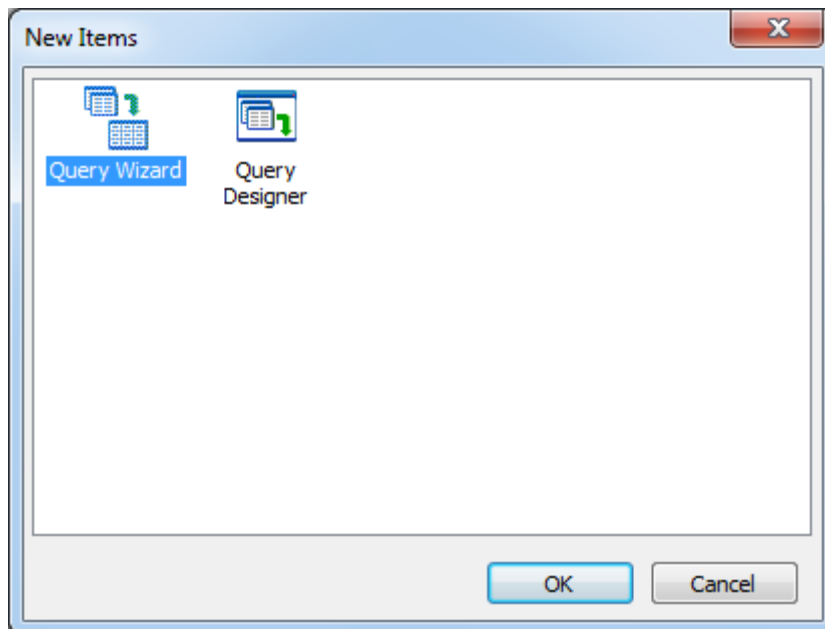
## Report Query Wizard

You can select data from your database using an SQL query. This functionality is provided via query-based dataviews, which can be visually created using the **Query Wizard** and visually maintained using the Query Designer.

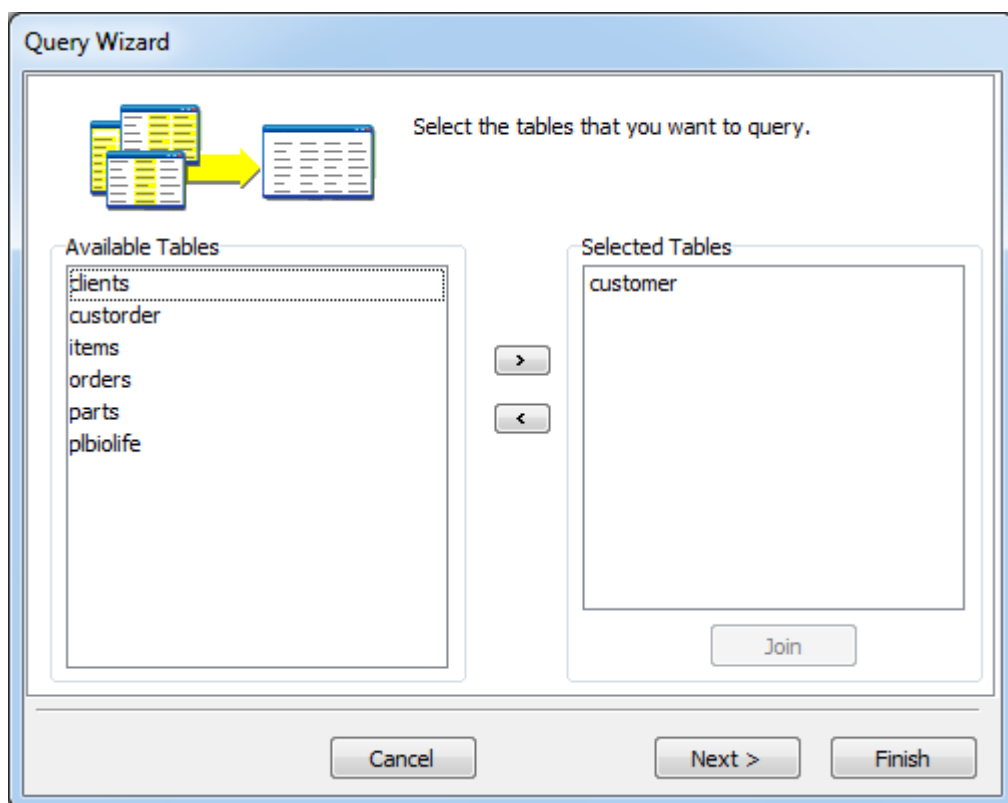
### Create a Simple Query-Based Dataview

The following series of screenshots shows how to create a simple query-based dataview via the Query Wizard.

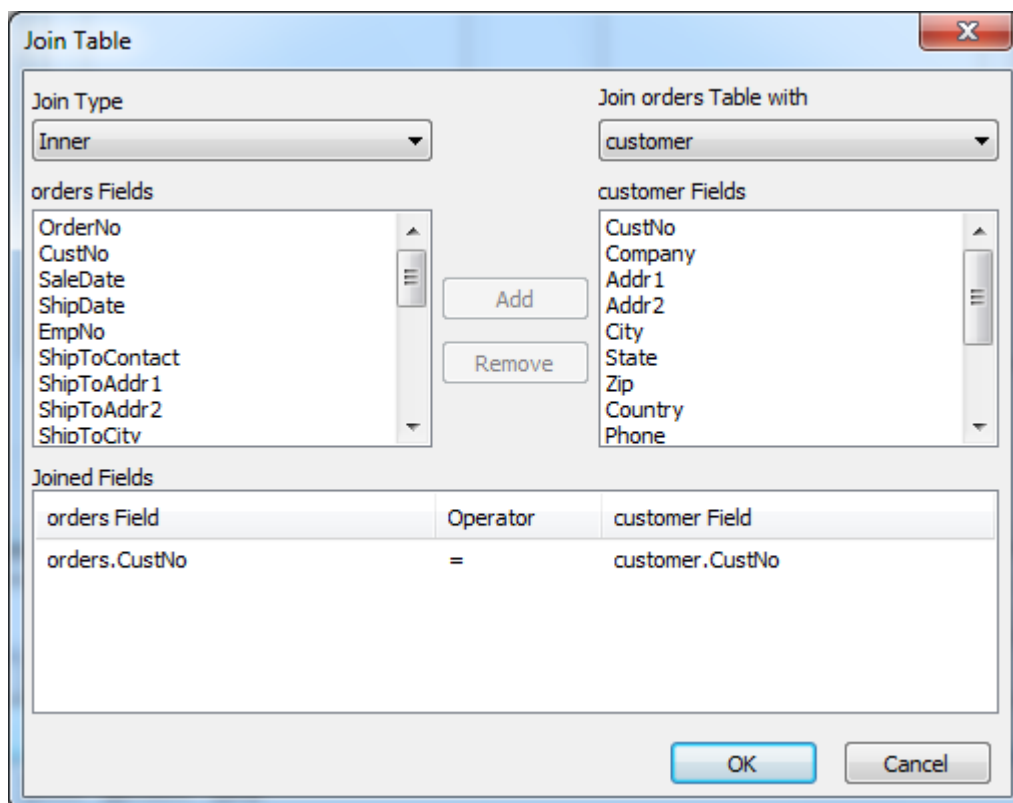
1. **Select File -> New... under Data tab. The New Dialog will be displayed. Double-click the Query Wizard icon.**



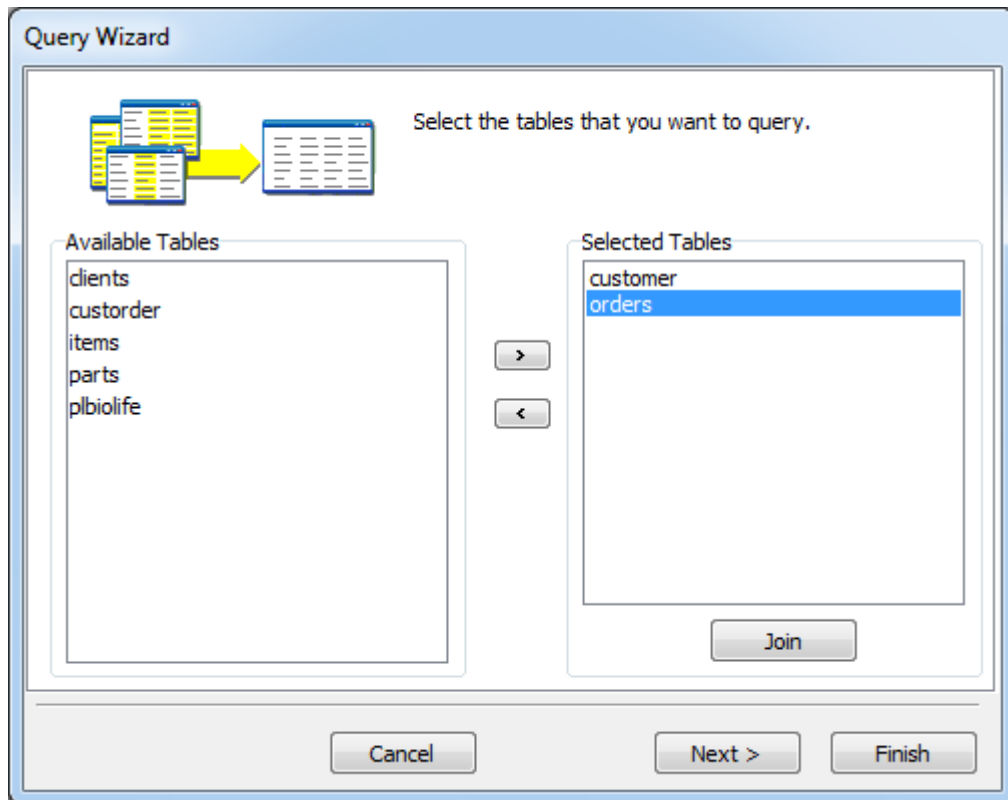
2. Select the first table for the query. The customer table has been selected.



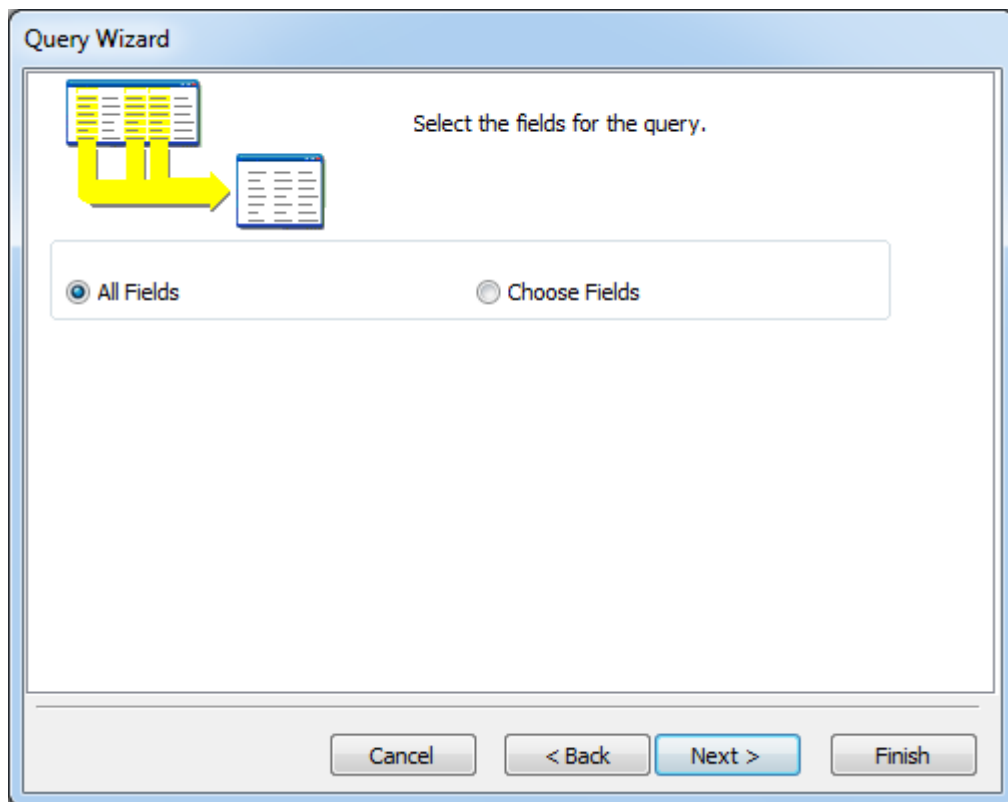
3. Select the second table for the query. The orders table has been selected and the Join dialog was automatically displayed. The dialog already contains the correct field linking (CustNo), so all we have to do is click OK.



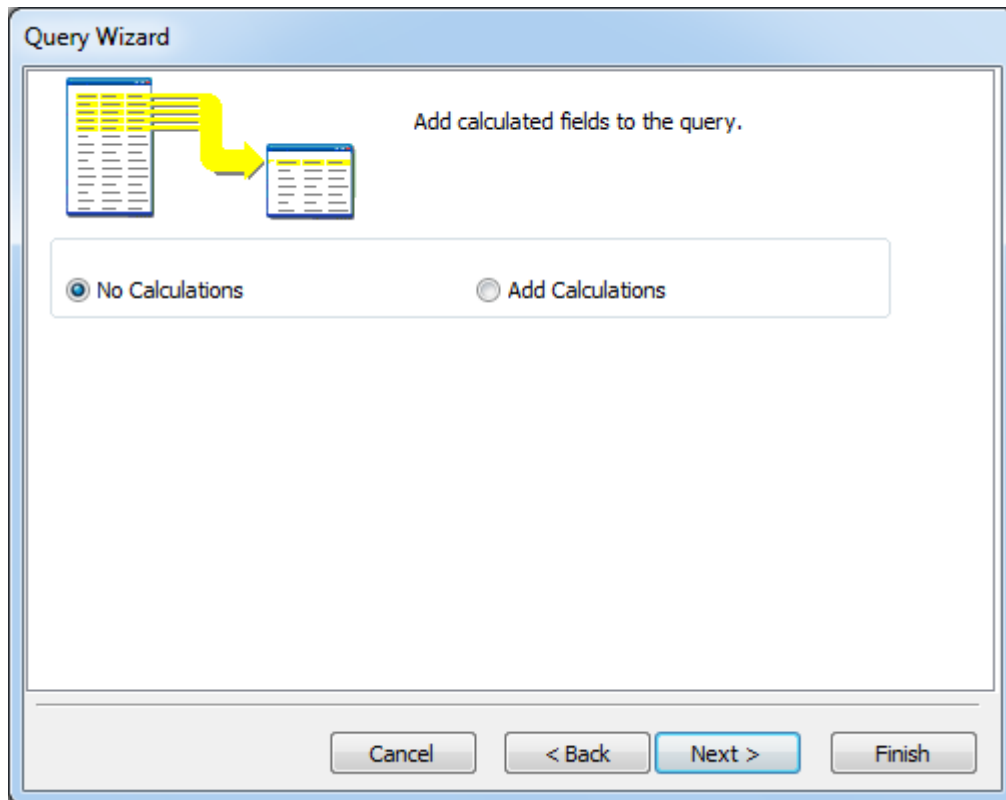
4. When we return to the query wizard, both tables are shown as selected.



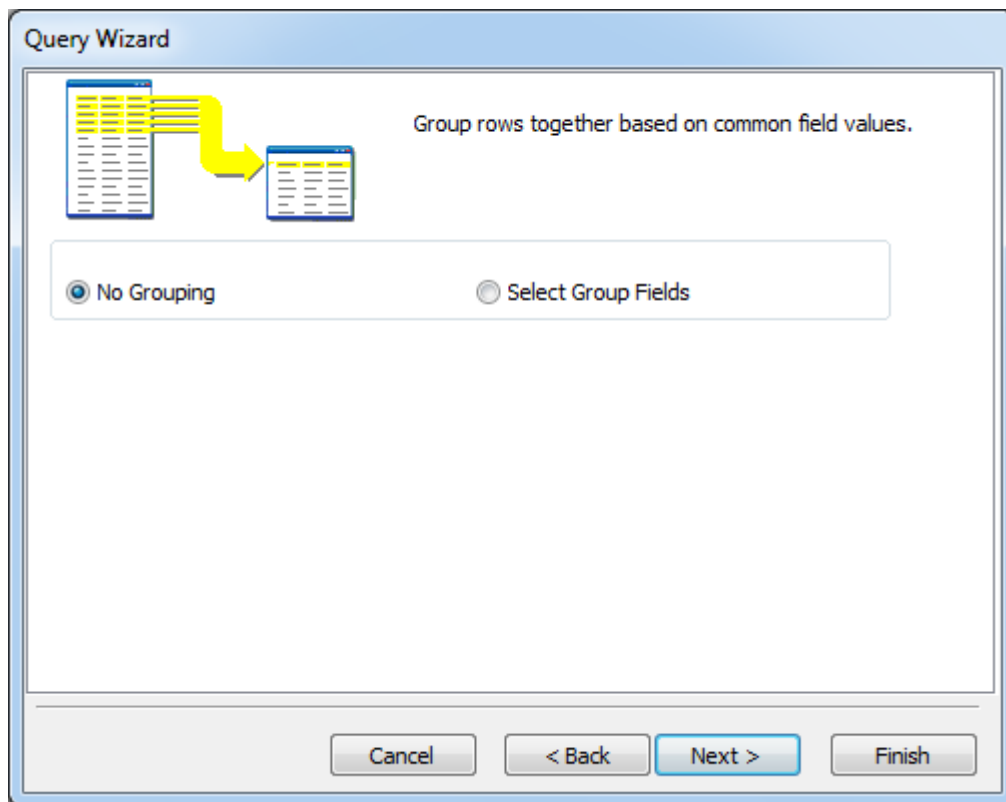
5. Skip the fields page, since we want to select all fields.



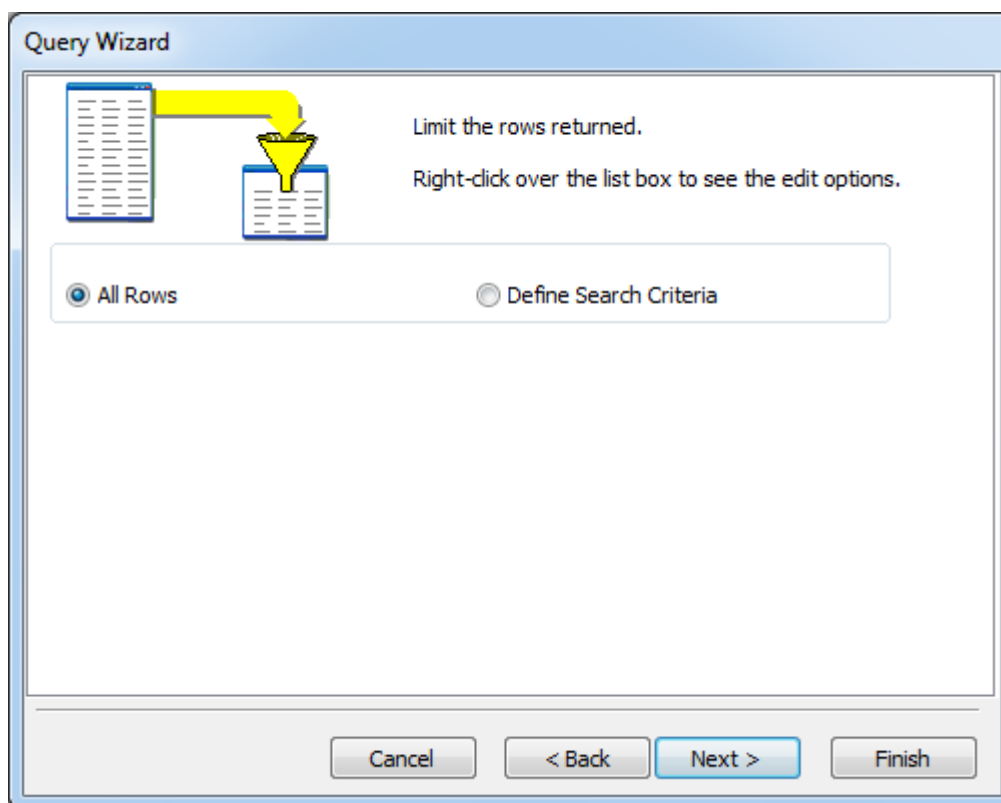
6. Skip the calculations page, since this query will not contain calculations.



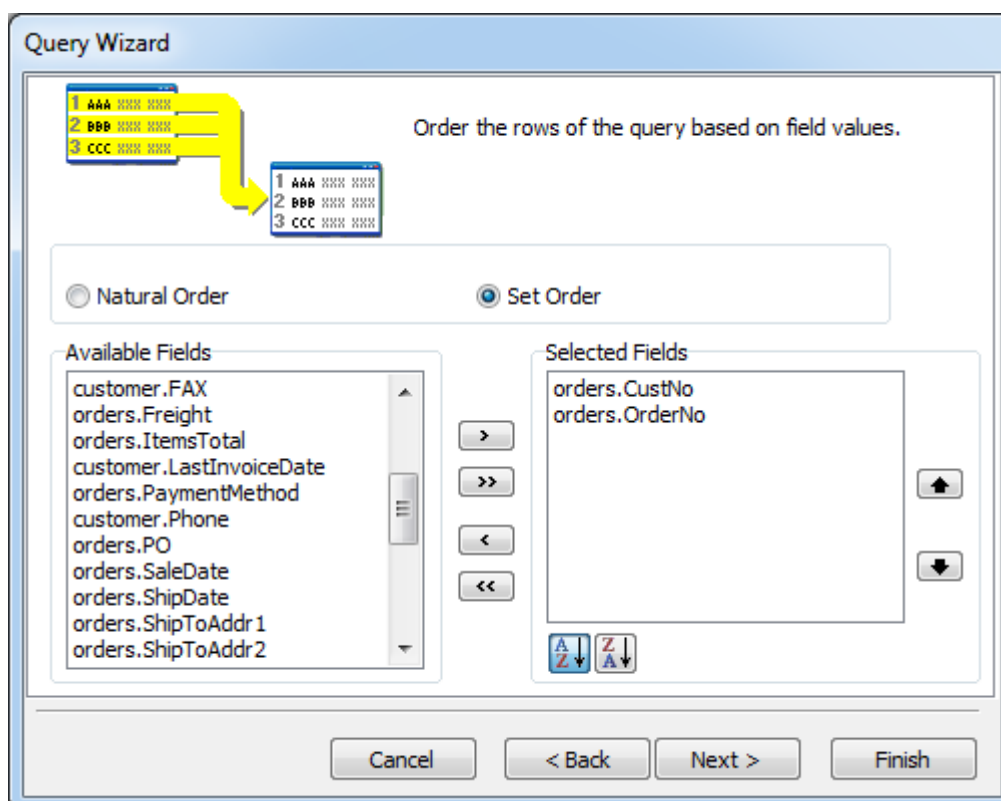
7. Skip the groups page, since this query will not be grouped.



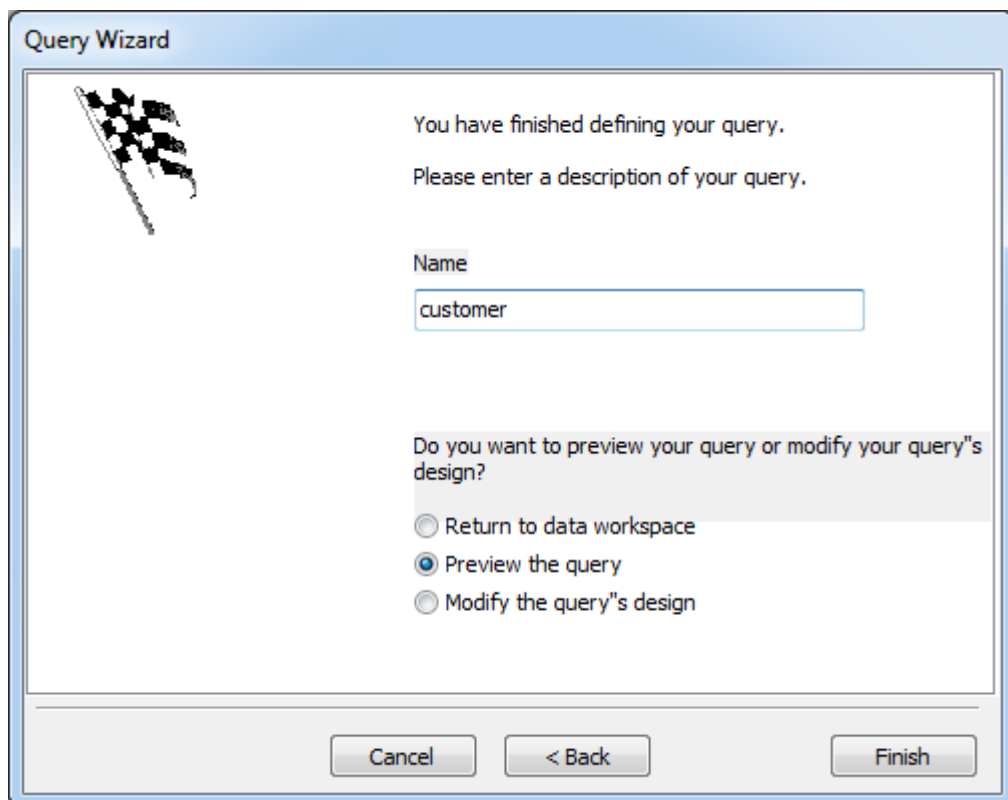
8. Skip the search criteria page; all records will be selected.



9. Set the order to CustNo, then OrderNo.



10. The query (data pipeline) name is automatically generated. The next action will be to preview the query.

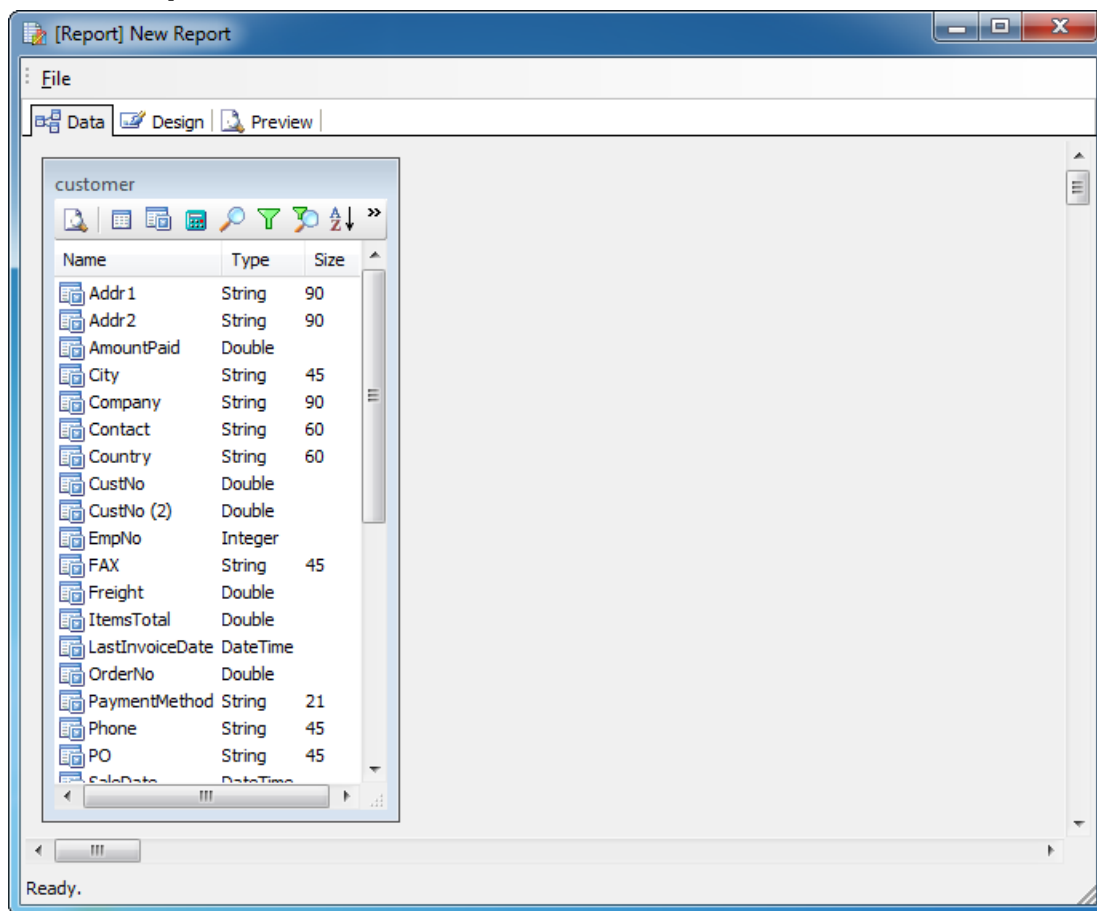


11. View the data to make sure the correct records have been selected.

CustNo (2)	EmpNo	FAX	Freight	ItemsTotal
1645	72	813-870-0282	0	101
1651	71	011-3-697043	0	20321.
1651	11	011-3-697043	0	343
1680	65	713-423-5676	0	26
1680	24	713-423-5676	0	185
1984	28	011-34-09064	0	101
1984	12	011-34-09064	0	31
2118	118	612-897-0348	0	52
2118	127	612-897-0348	0	126
2135	114	503-555-2769	0	206
2156	145	803-509-0553	0	78
2156	61	803-509-0553	0	99
2163	52	011-32-09485	0	37
2163	46	011-32-09485	0	10064.
2163	144	011-32-09485	0	7
2165	2	011-32-44938	0	3
2315	24	30-661-05943	0	766

Record 1 of 51

12. Close the preview window. The dataview is then created.

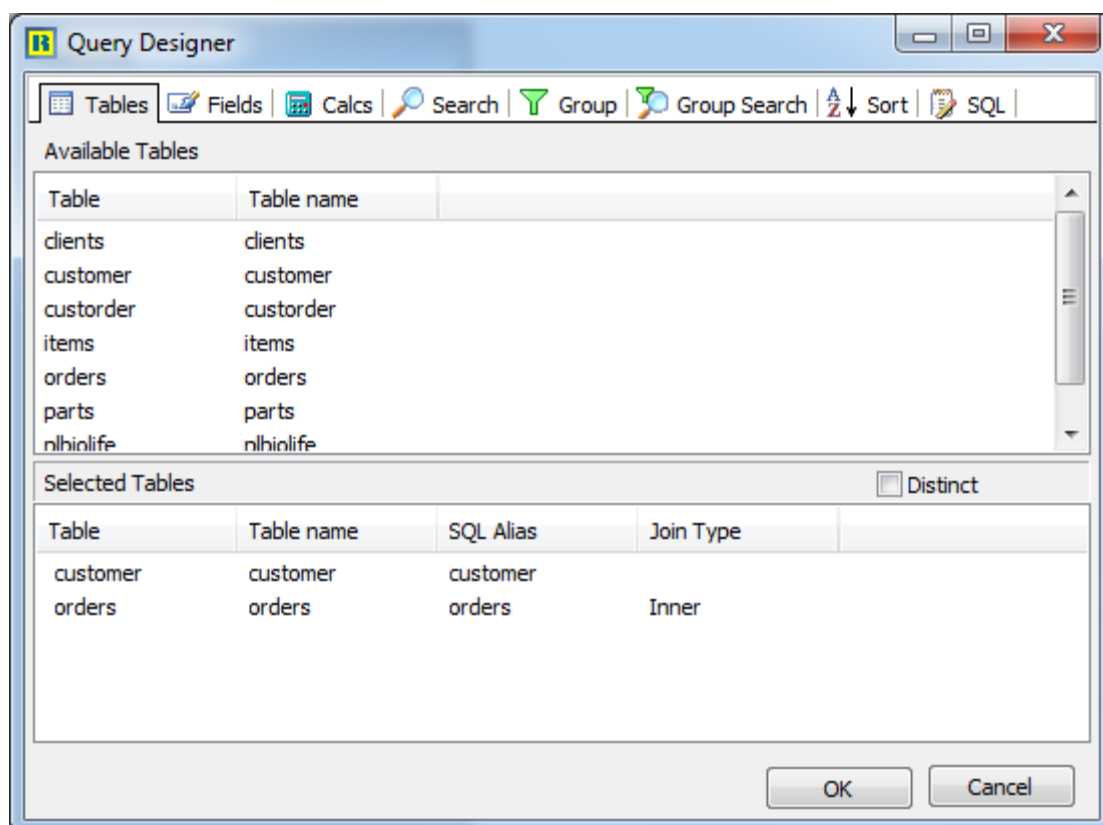




## Report Query Designer

The **Query Designer** is used to modify query-based dataviews created by Query Wizard. The Query Designer presents a series of notebook tabs; each tab represents a different part of the query.

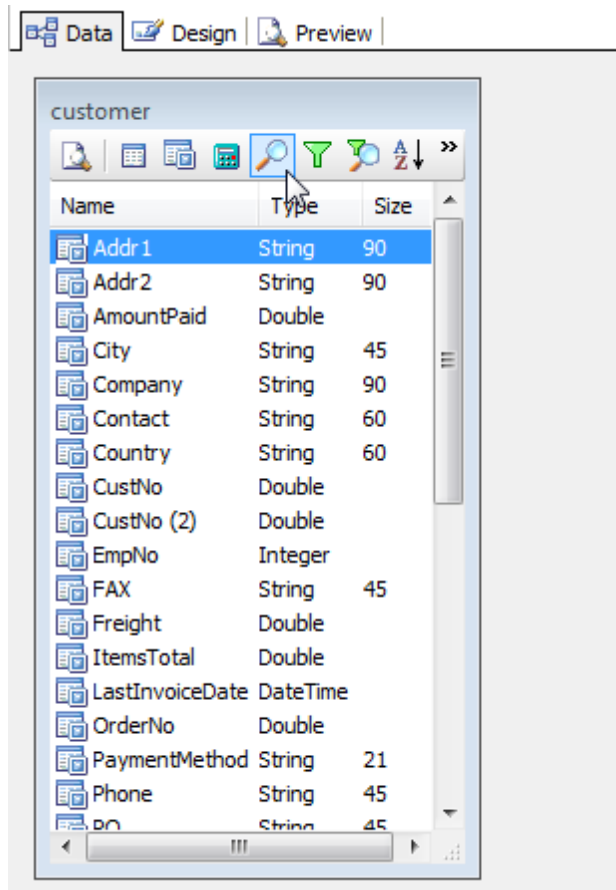
- [Add Search Criteria](#)
- [Create a Group Sum](#)
- [Concatenate Fields](#)
- [Edit SQL](#)



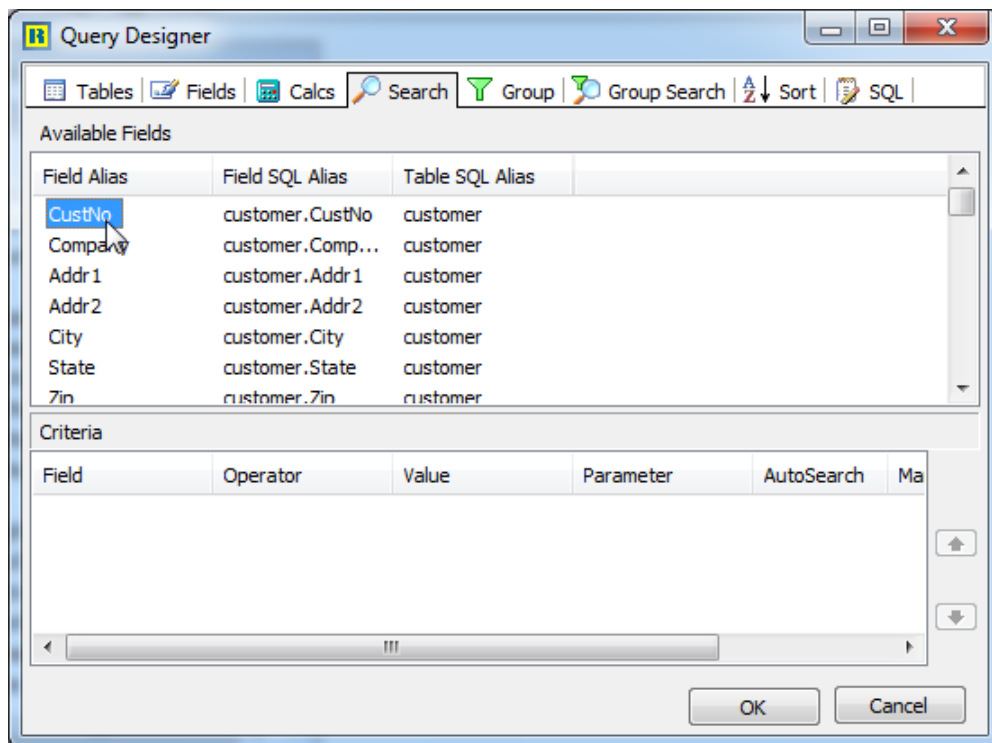
## Add Search Criteria

You can use the Query Designer to add or remove search criteria from your query. Perform these steps in order to add search criteria:

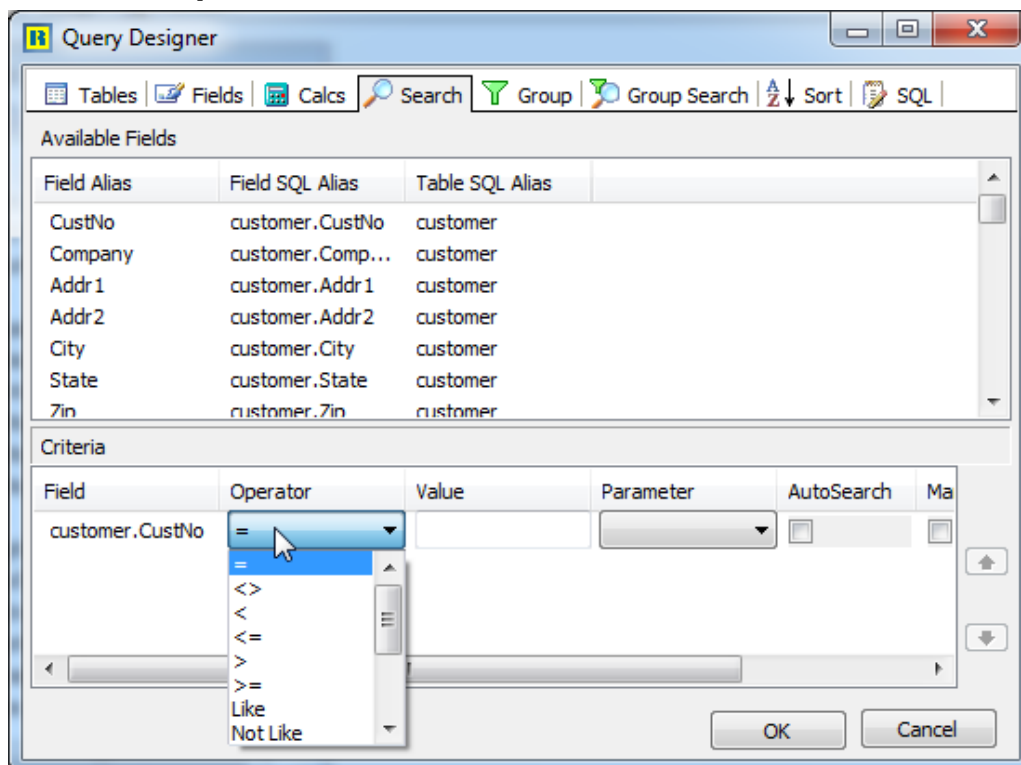
1. **Click on the Search icon of the dataview to launch the Query Designer.**



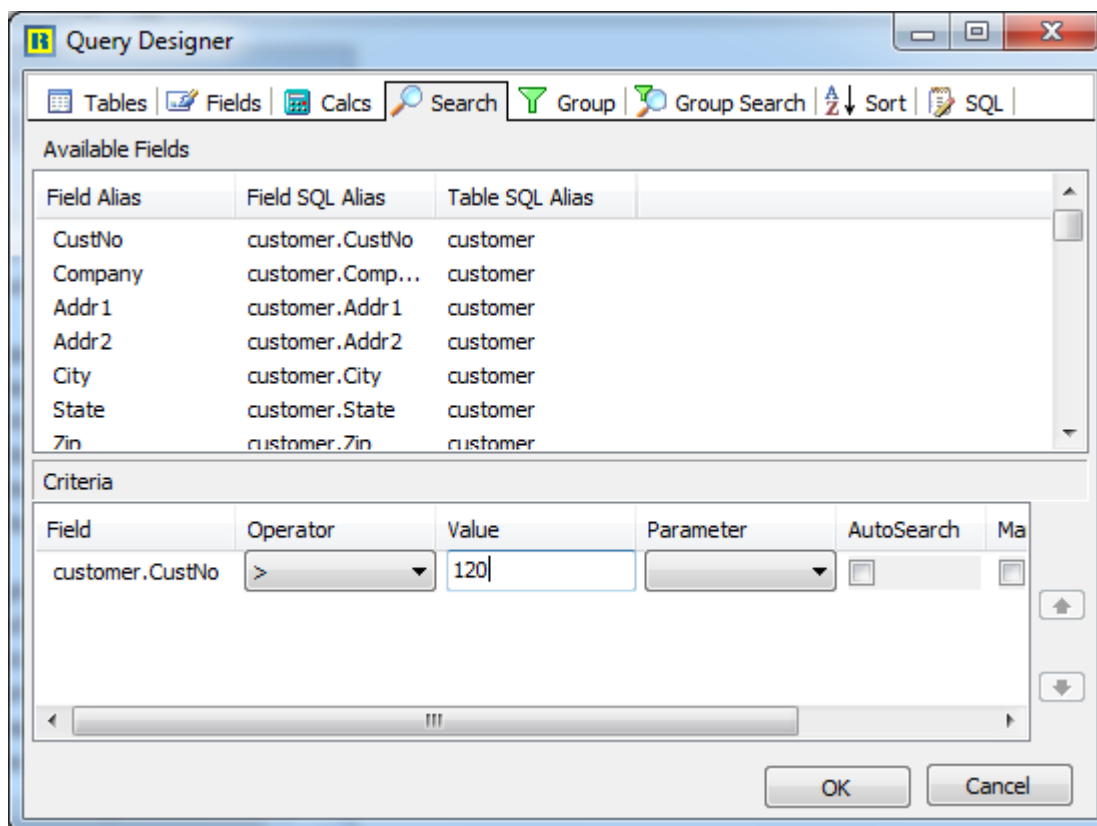
2. From the list of fields at the top of the search page, double-click on the field for which criteria needs to be entered.



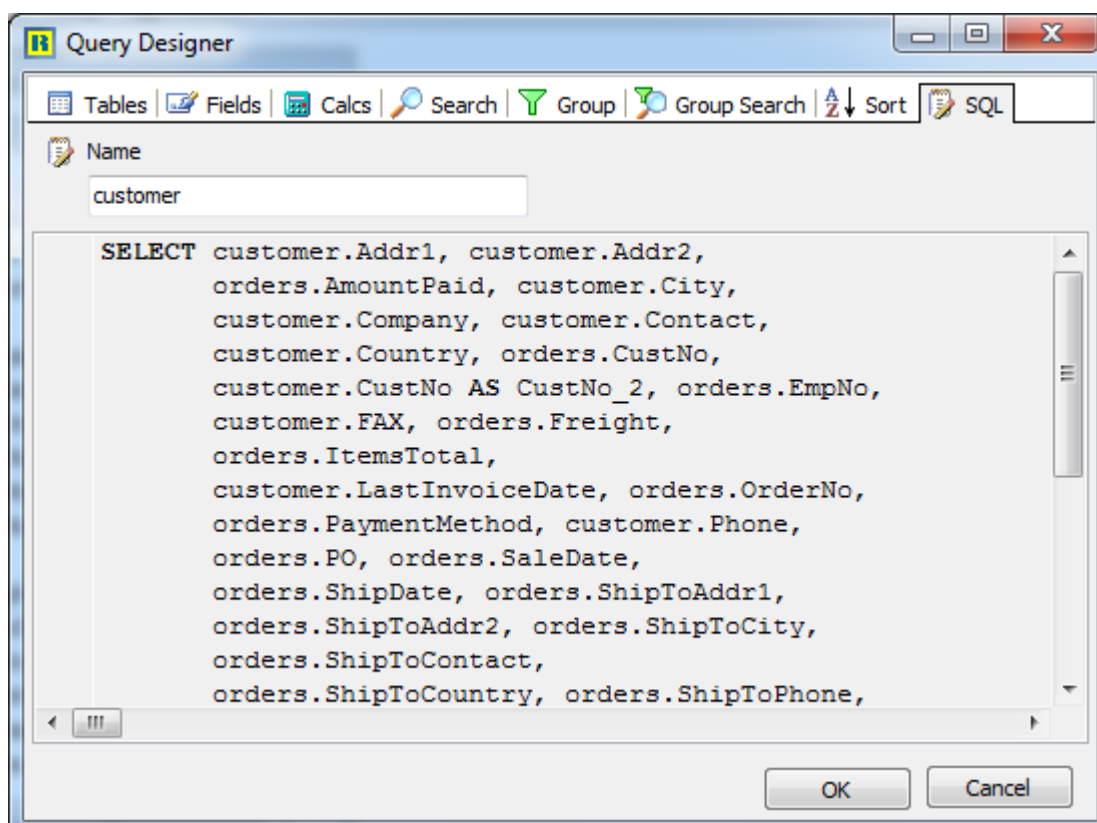
3. Click on the field that has been added to the list of criteria at the bottom and select the operator.



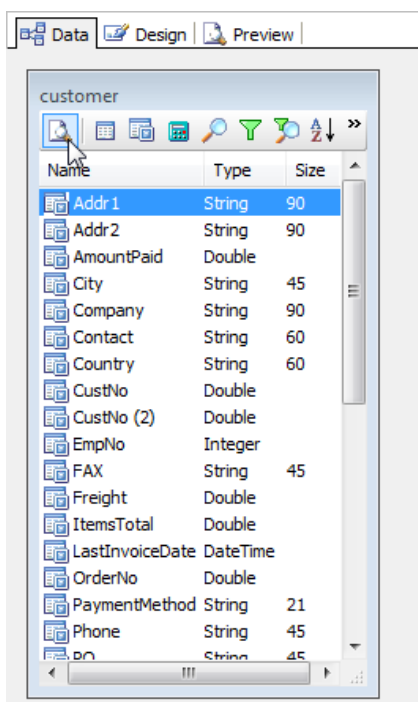
- Click in the edit box and enter the search criteria value. This criteria will find all customer id that are greater than 120.



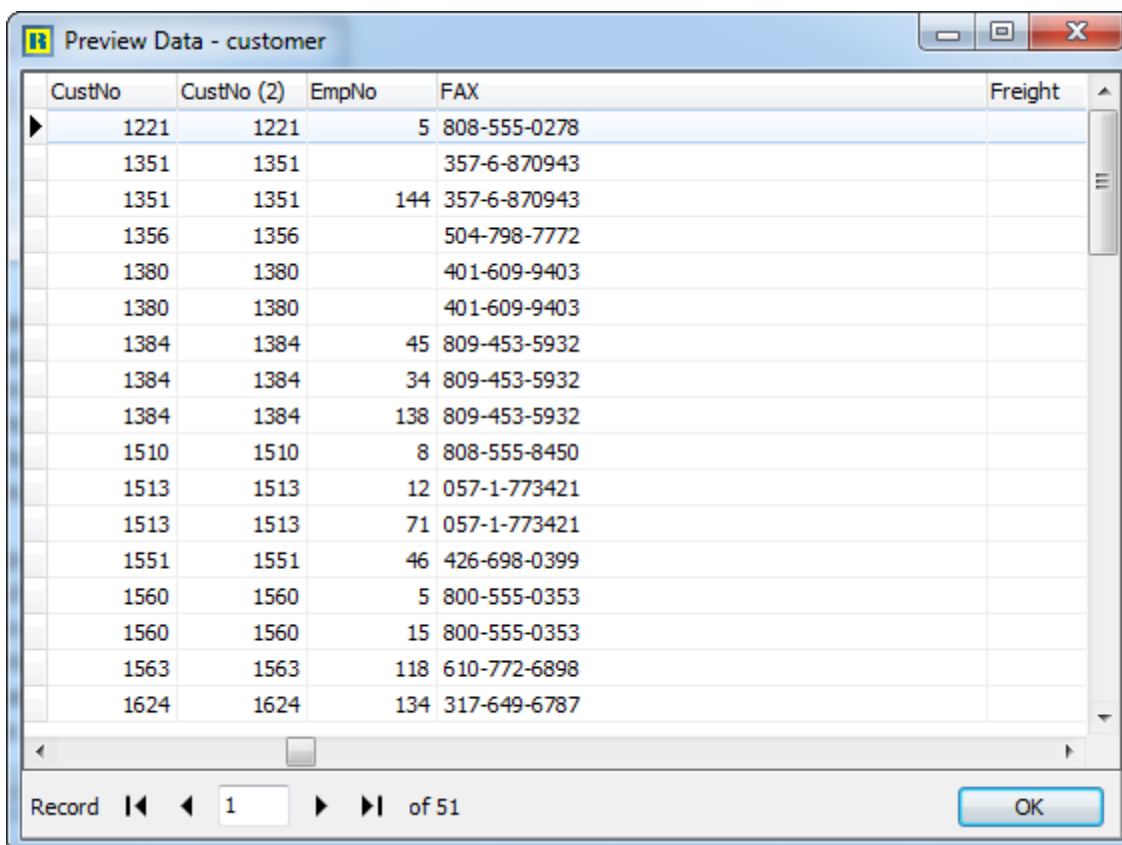
- Click on the SQL tab to make sure the criteria value is valid.



## 6. Close the Query Designer and click on the Preview icon.



## 7. Preview the data and make sure that the intended records are selected.



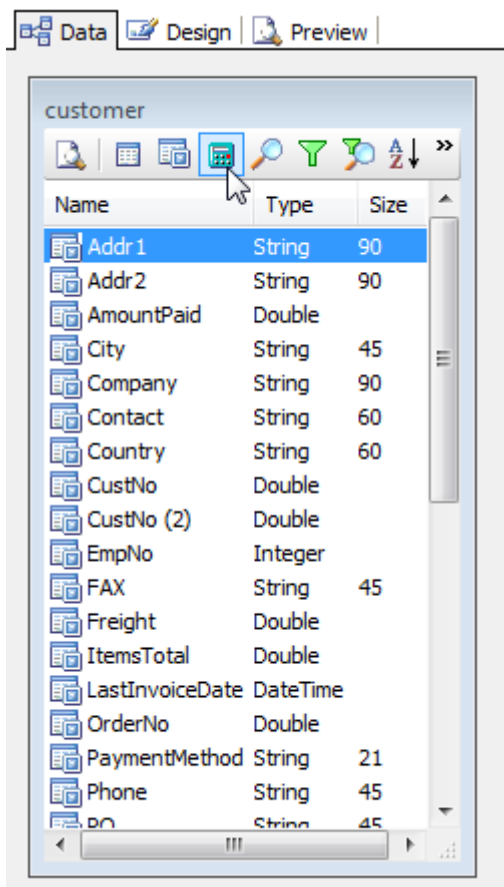
## Create a Group Sum

The SQL 'GROUP BY' clause allows you to eliminate rows in your query where the field values repeat. For example, let's assume we have a database table that contains order records. Each order record has the customer number and the amount paid. If we viewed the data in this table, we would see that the value in the customer number field repeats where there are multiple orders for a customer.

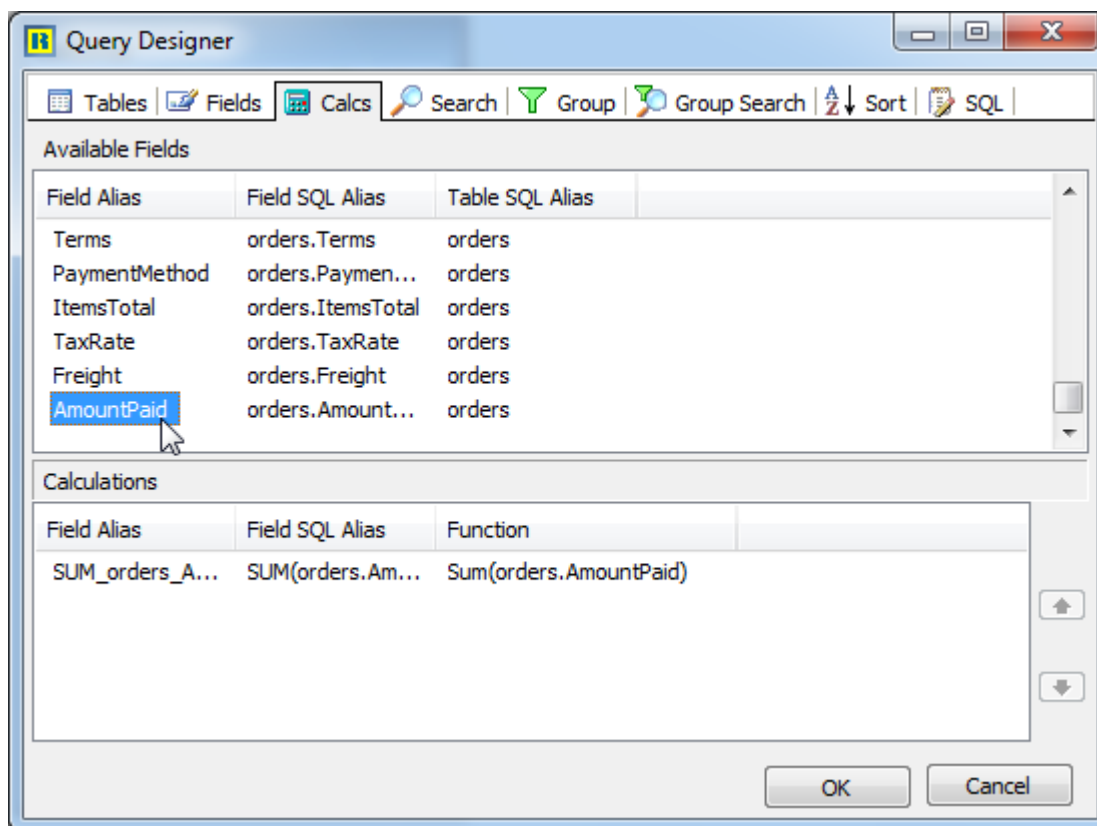
We can use SQL to select data from the orders table and calculate the total amount paid for each customer. We can do this by specifying a group on the customer number field. By specifying the group, we are saying to the SQL engine: create one row in the result set for each customer number found. When the SQL engine runs the query, it will find multiple records for some customers; these records will be eliminated from the result set. SQL allows us to perform calculations on these repeated records and store the result in a new field of the result set.

These types of calculations can be created on the Calc tab of the Query Designer. Perform these steps in order to sum the amount paid for all customers in the orders table:

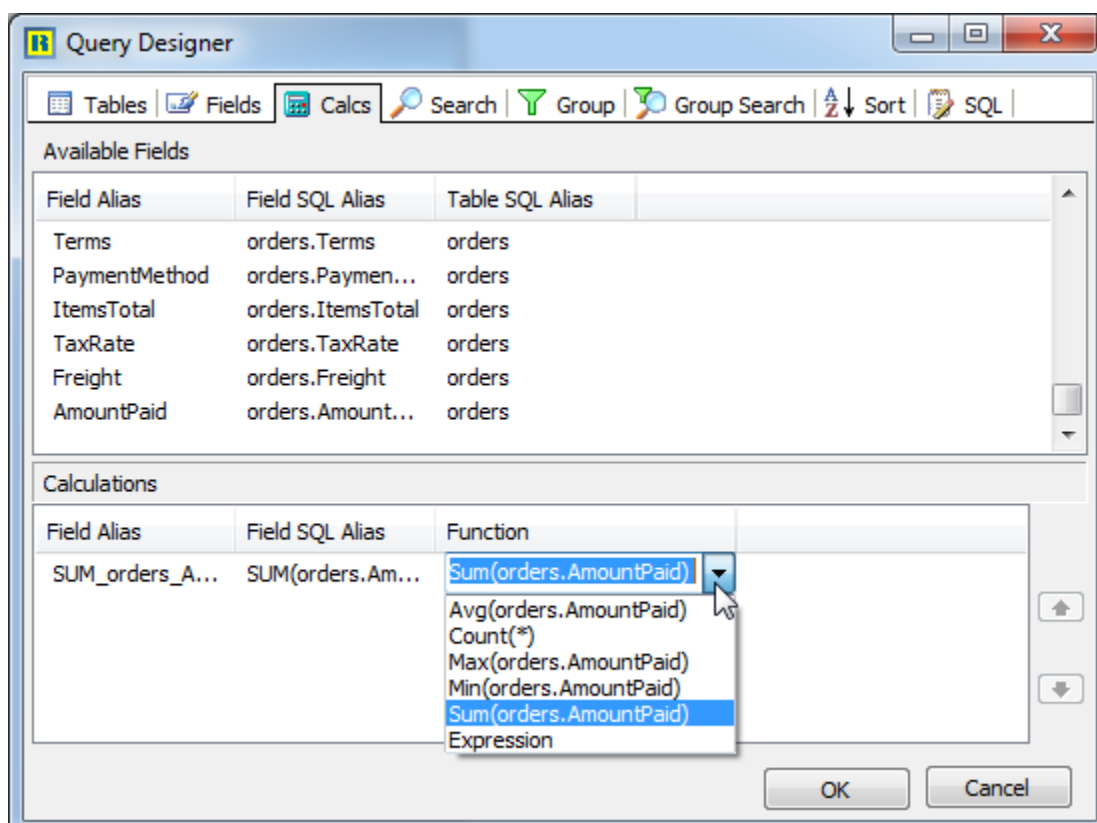
1. **Click the 'Calc' icon to launch the Query Designer.**



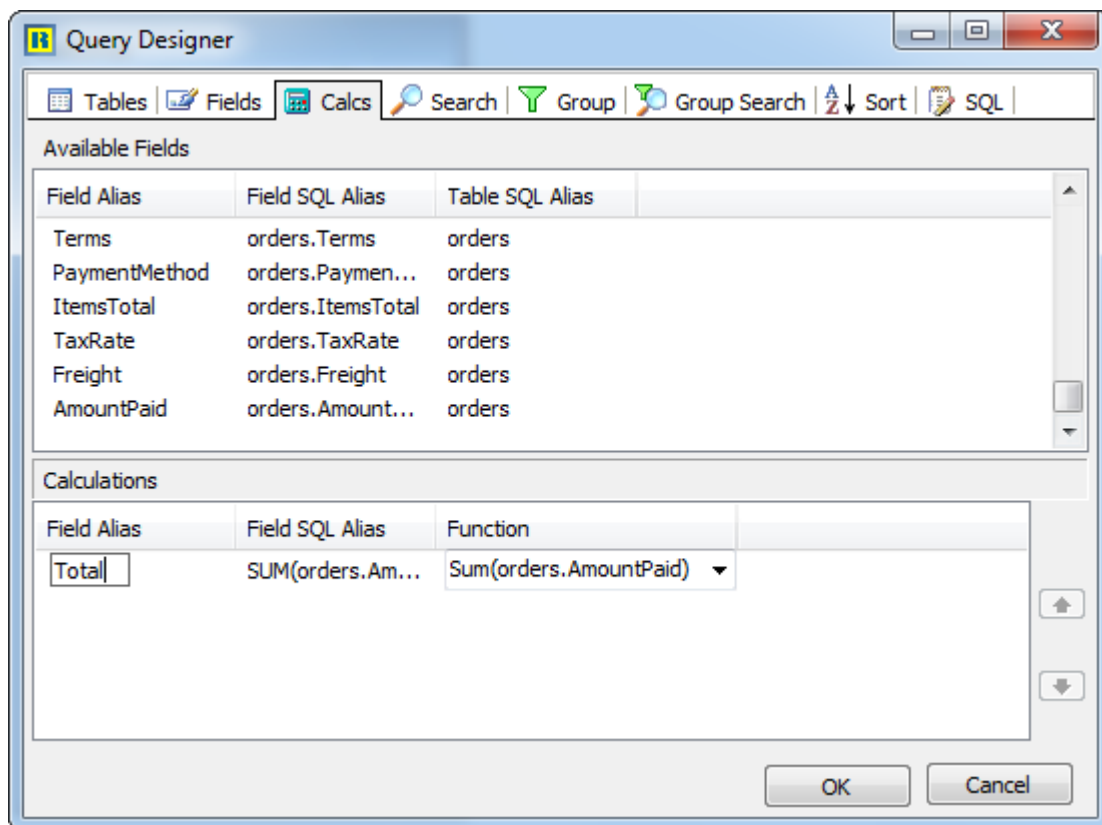
2. Double-click the 'Amount Paid' field from the selection list at the top of the page. Amount Paid will be added to the list of calculations.



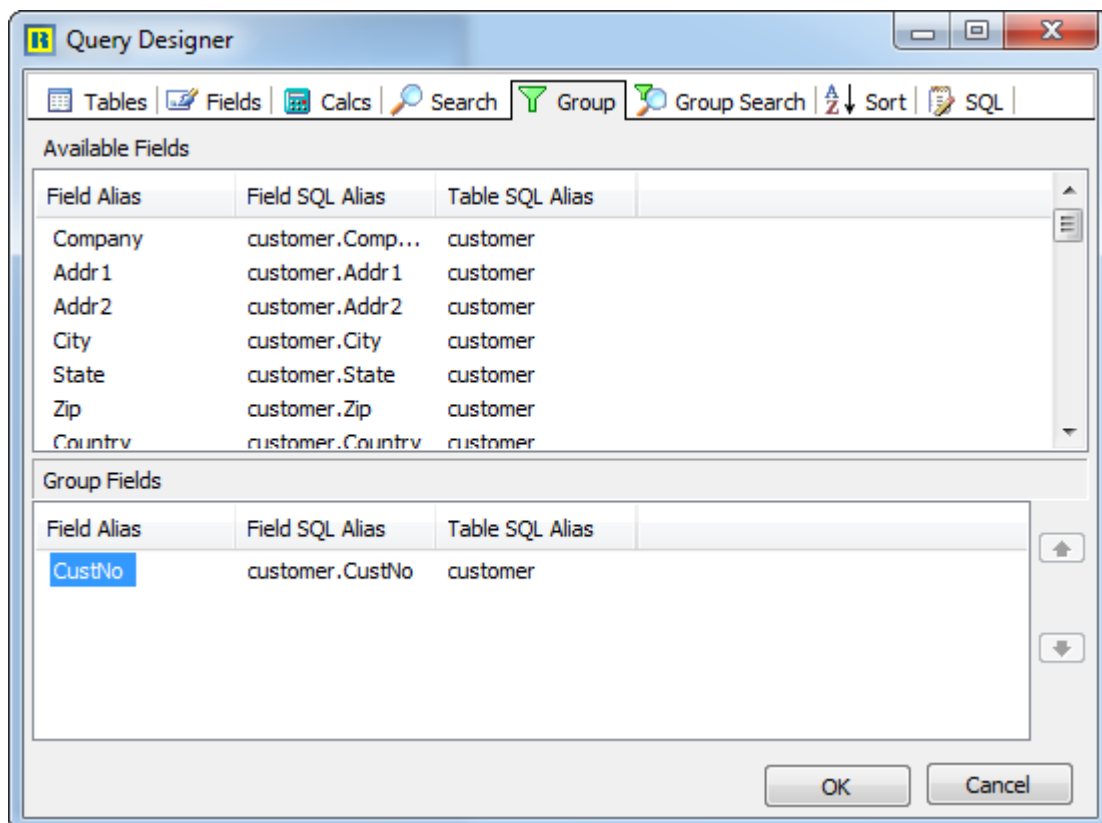
3. Select 'Sum' as the function type for the calculation.



4. Enter the Field Alias you would like to use for this calculated field.

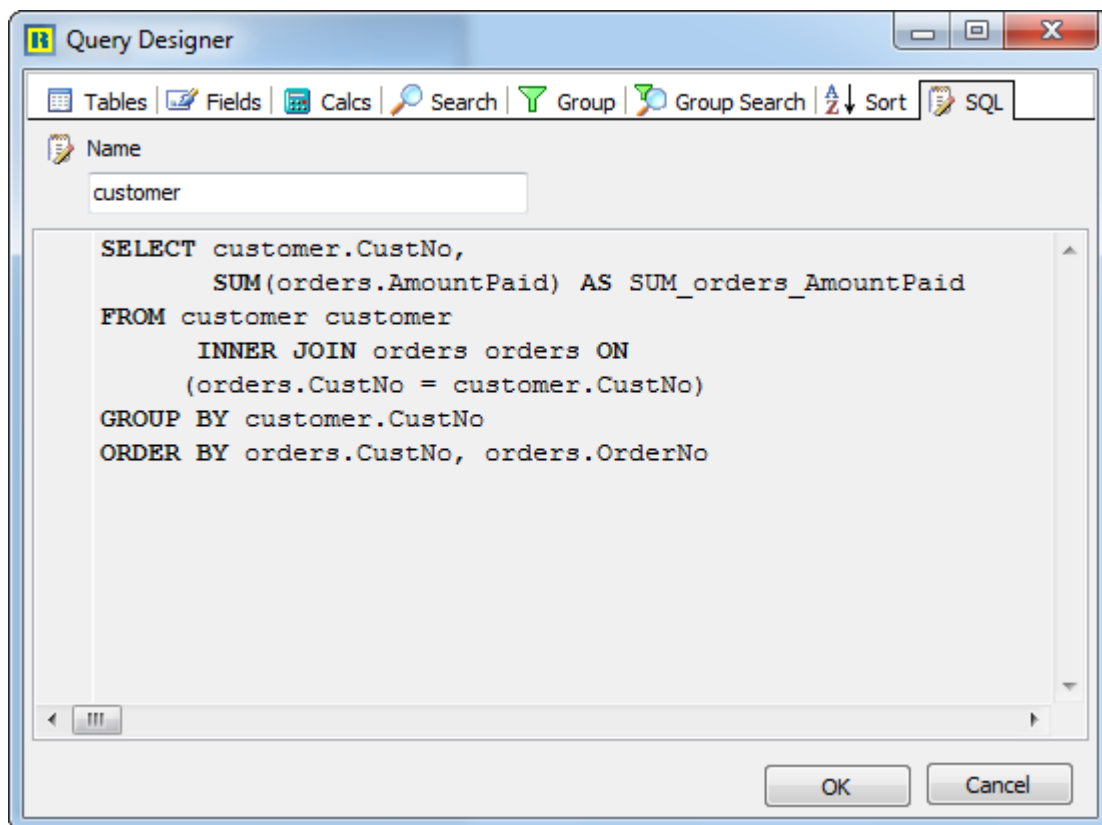


5. Click the Group Tab, check the grouping.

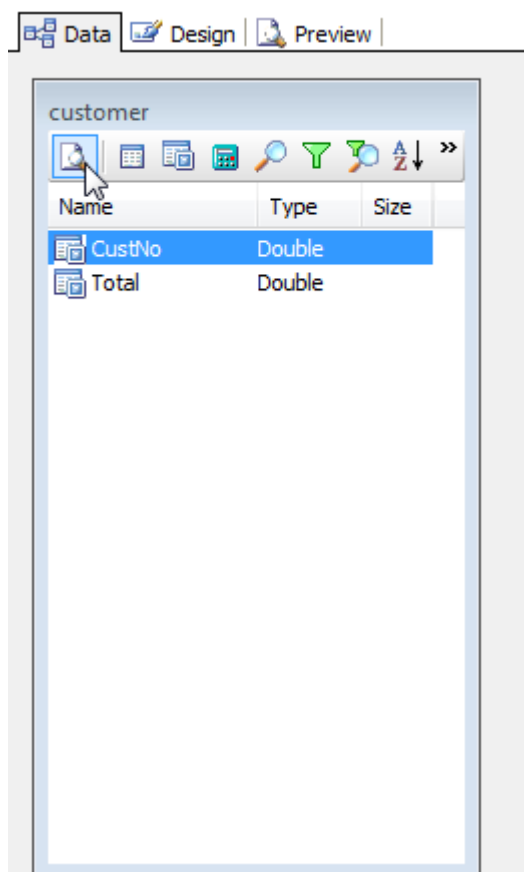




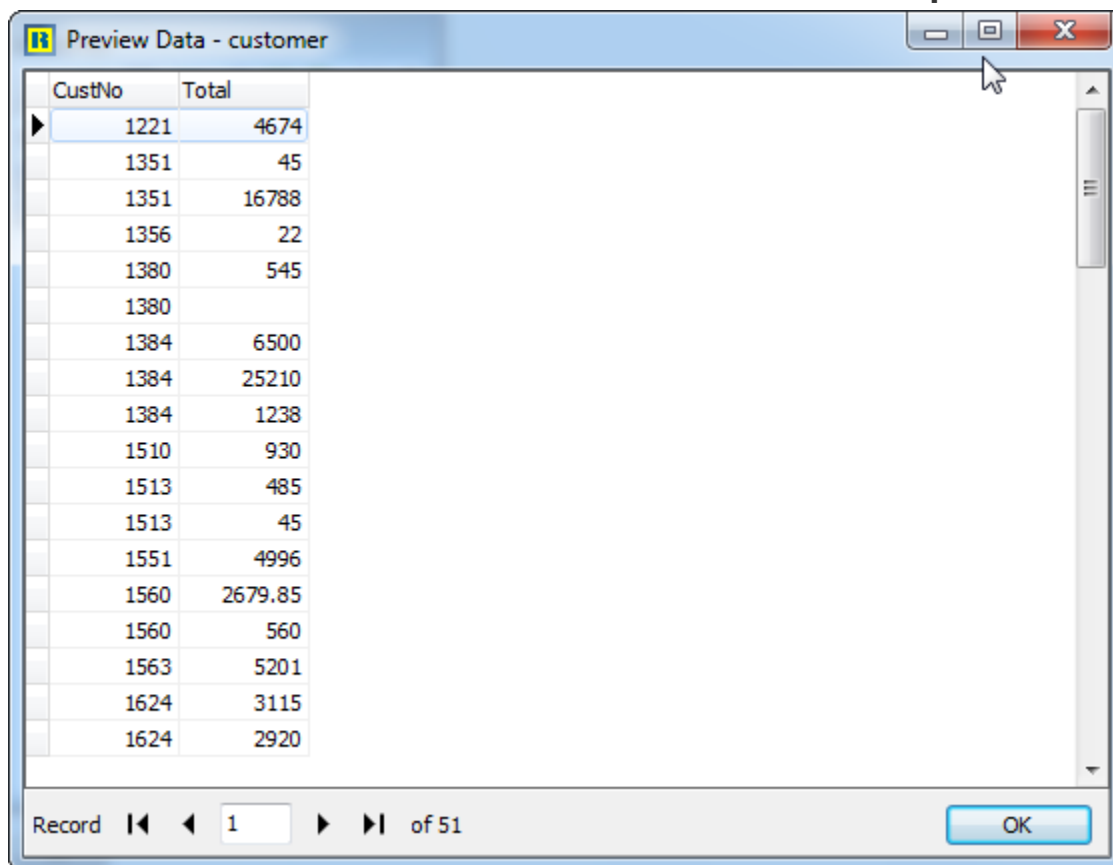
- Click the SQL tab to make sure the generated SQL is valid.



- Close the Query Designer and click the Preview icon to preview the data.



8. Check the data to make sure the sum is calculated as expected.

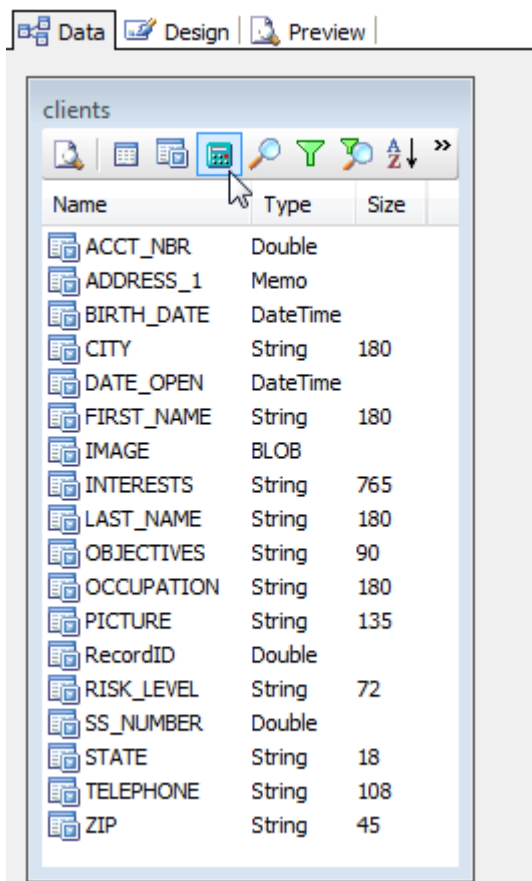


CustNo	Total
1221	4674
1351	45
1351	16788
1356	22
1380	545
1380	
1384	6500
1384	25210
1384	1238
1510	930
1513	485
1513	45
1551	4996
1560	2679.85
1560	560
1563	5201
1624	3115
1624	2920

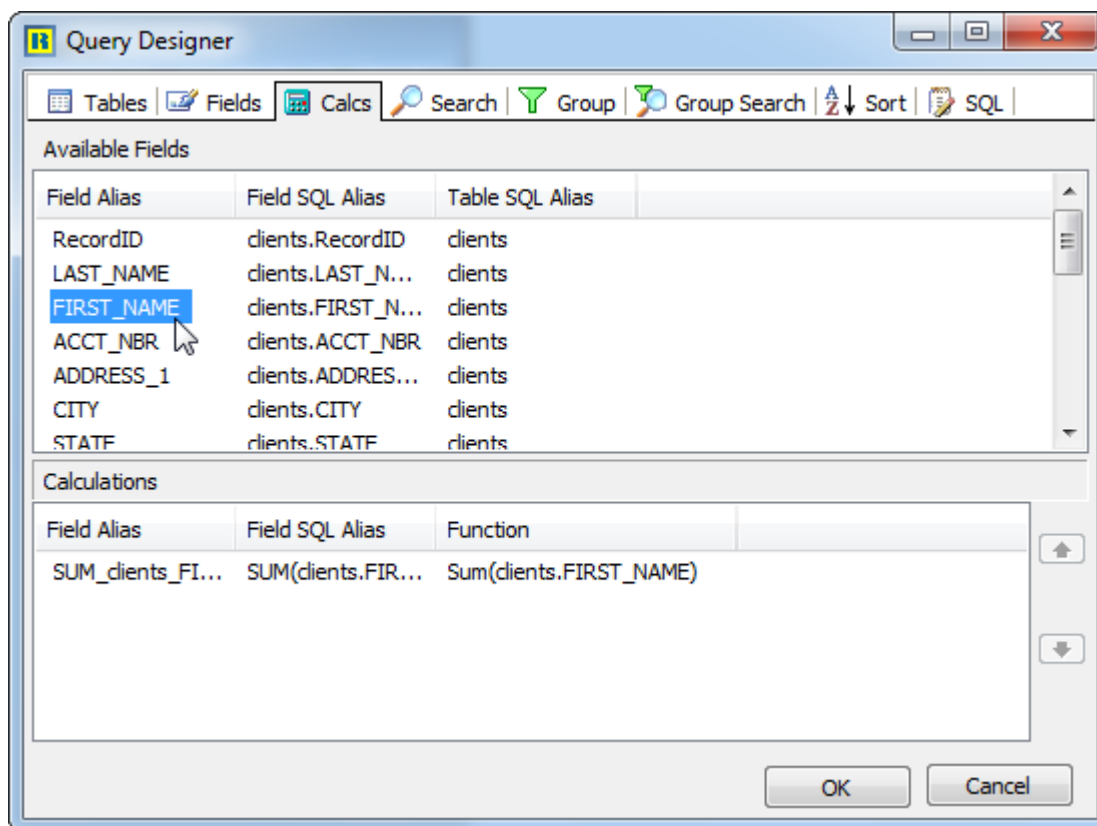
## Concatenate Fields

You can enter SQL expressions from the **Calcs** tab of the Query Designer. The following query selects data from a table of employees. The table has a first name and last name field. Perform these steps in order to concatenate these two fields together using the Query Designer:

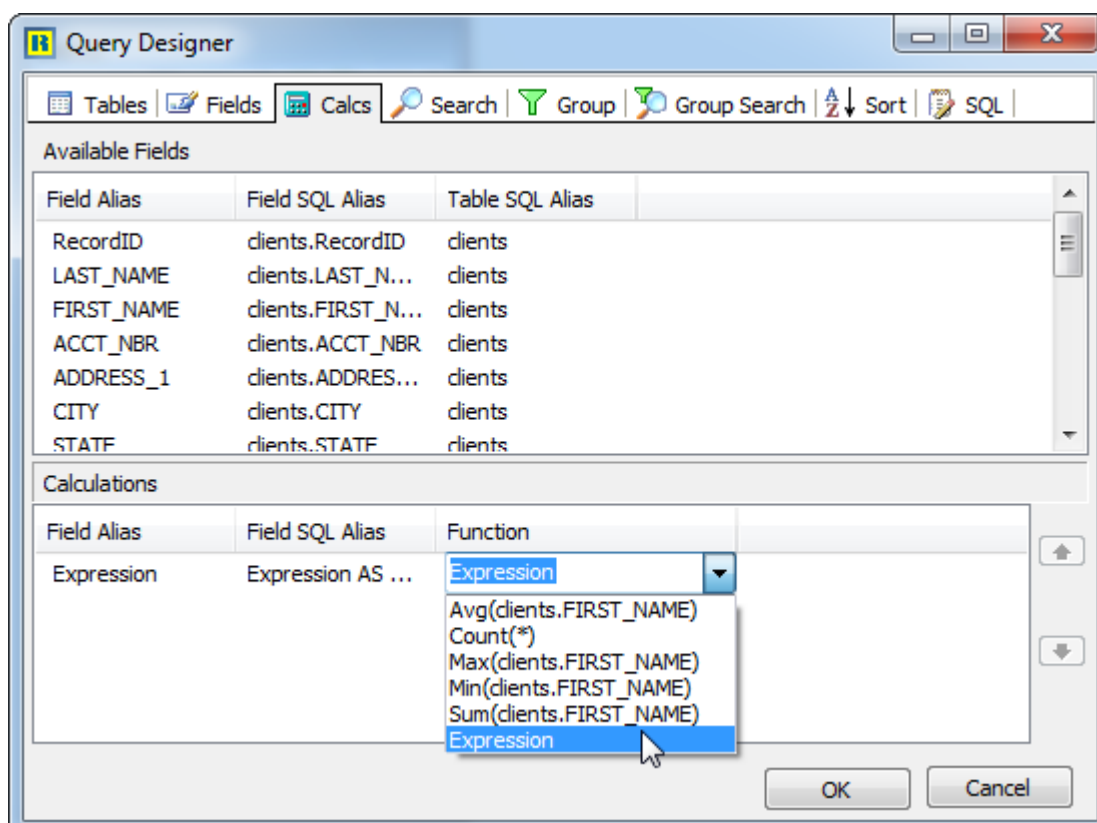
1. **Click the 'Calcs' icon to launch the Query Designer.**



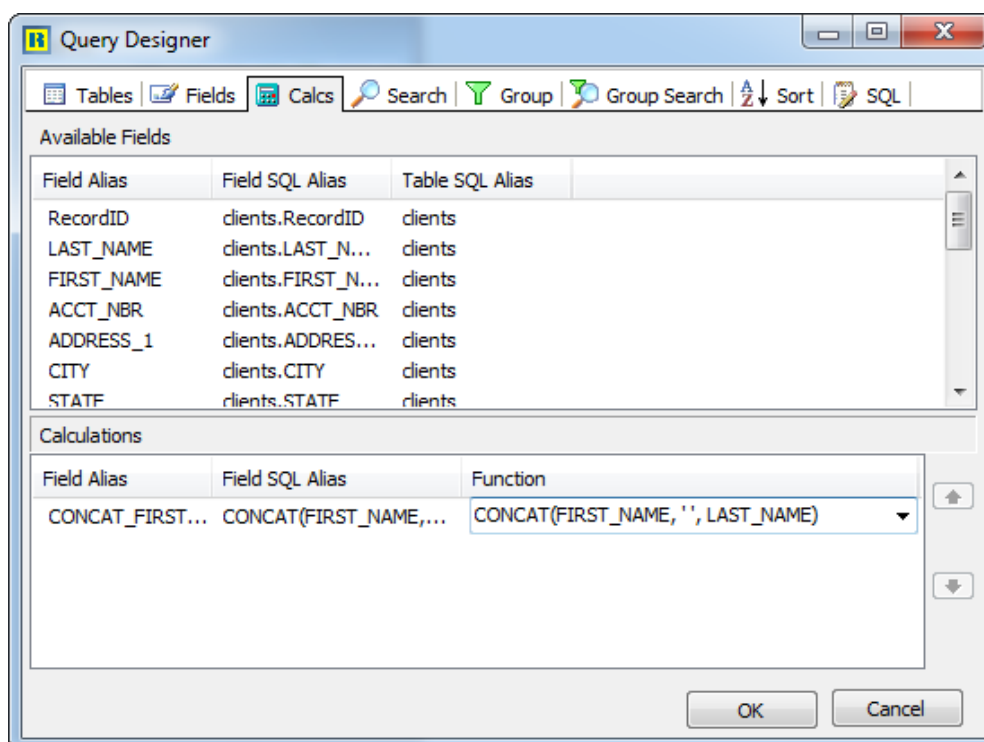
2. Double-click the 'First Name' field from the selection list at the top of the page. 'First Name' will be added to the list of calculations.



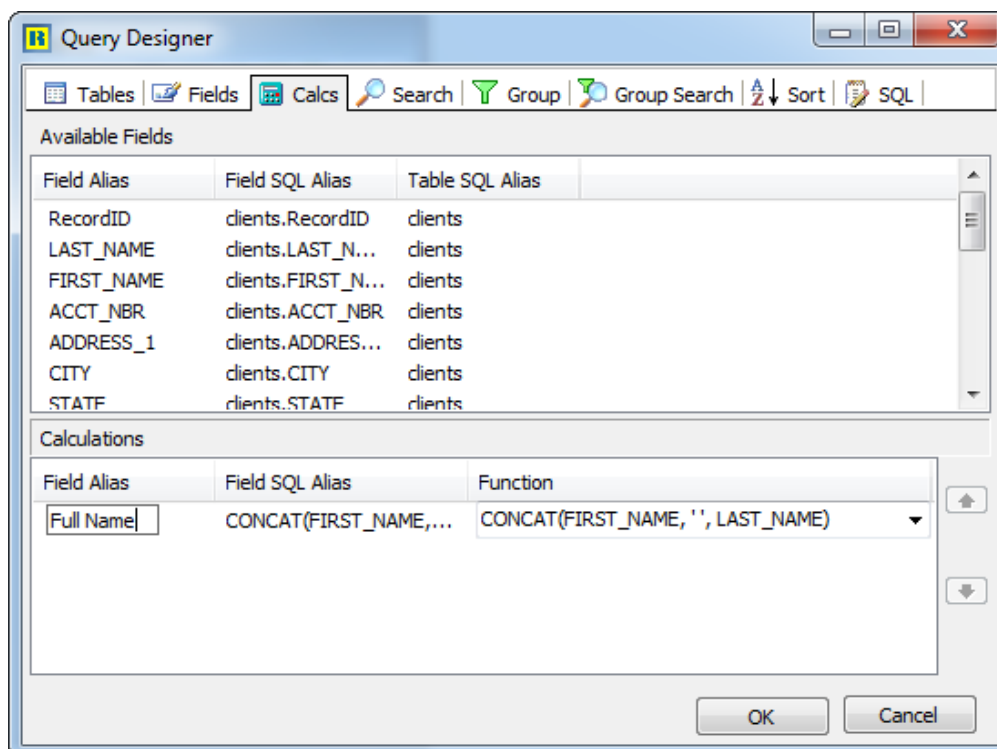
3. Select 'Expression' as the function type for the calculation.



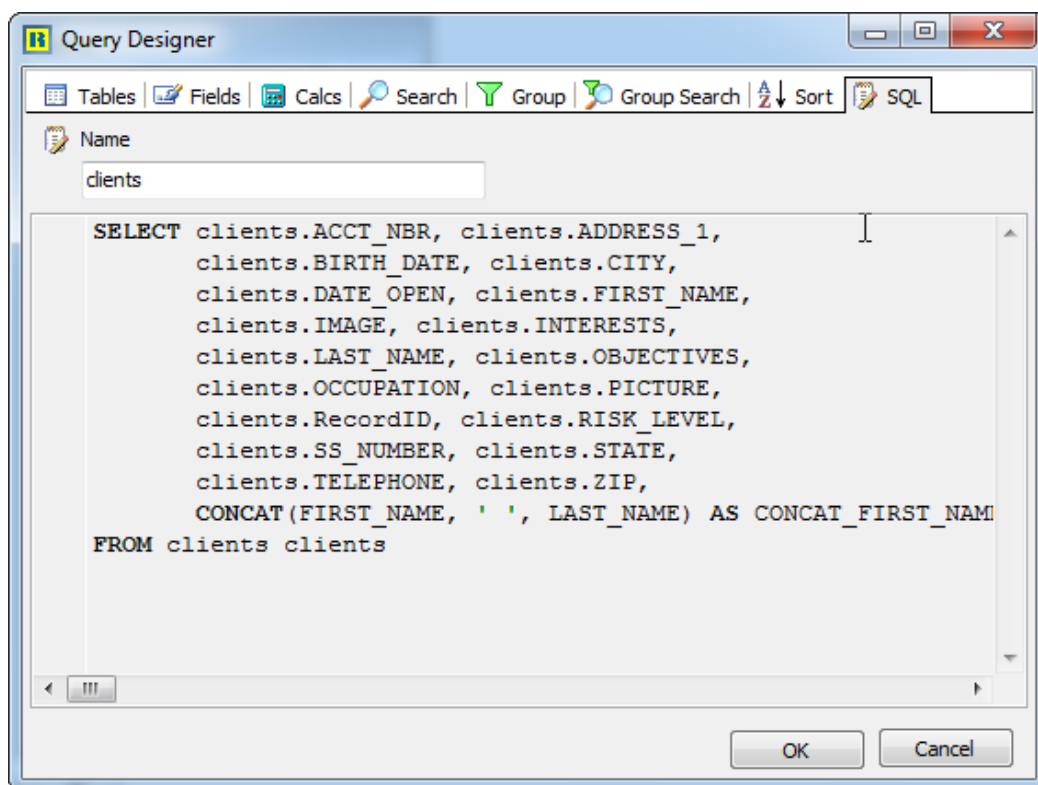
4. **Modify the widths of the 'Field SQL Alias' and 'Function' columns in the calculations list at the bottom of the page so that there is enough space to enter the expression. The figure below illustrates a sample expression.**



5. **Enter the Field Alias you would like to use for this calculated field.**



6. Click the SQL tab to make sure the generated SQL is valid.



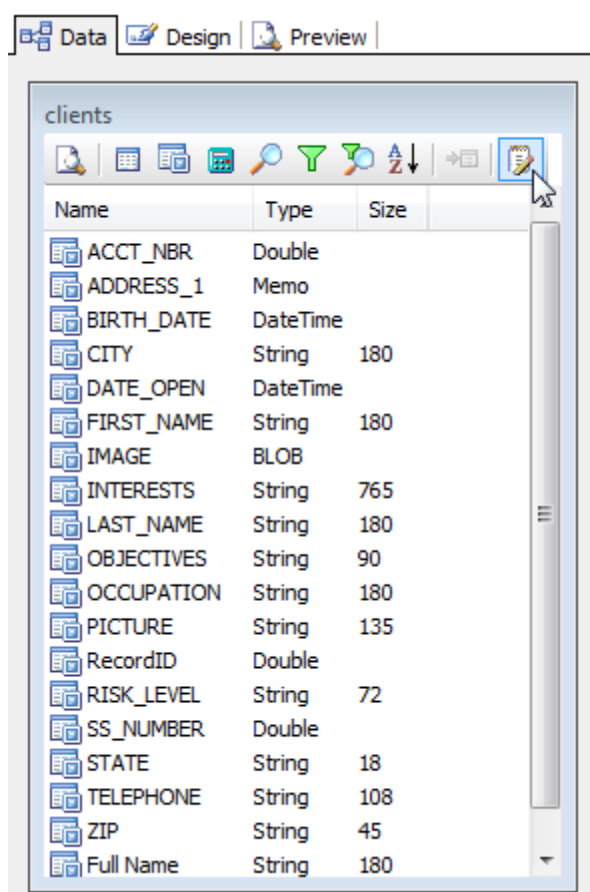
## Edit SQL

There may be times when you need to utilize advanced features of SQL that cannot be accessed via the visual interface of the Query Designer. In these cases, you can edit the SQL manually in the Query Designer. Once you have edited the SQL manually, you must always use the SQL tab of the Query Designer to make future modifications.

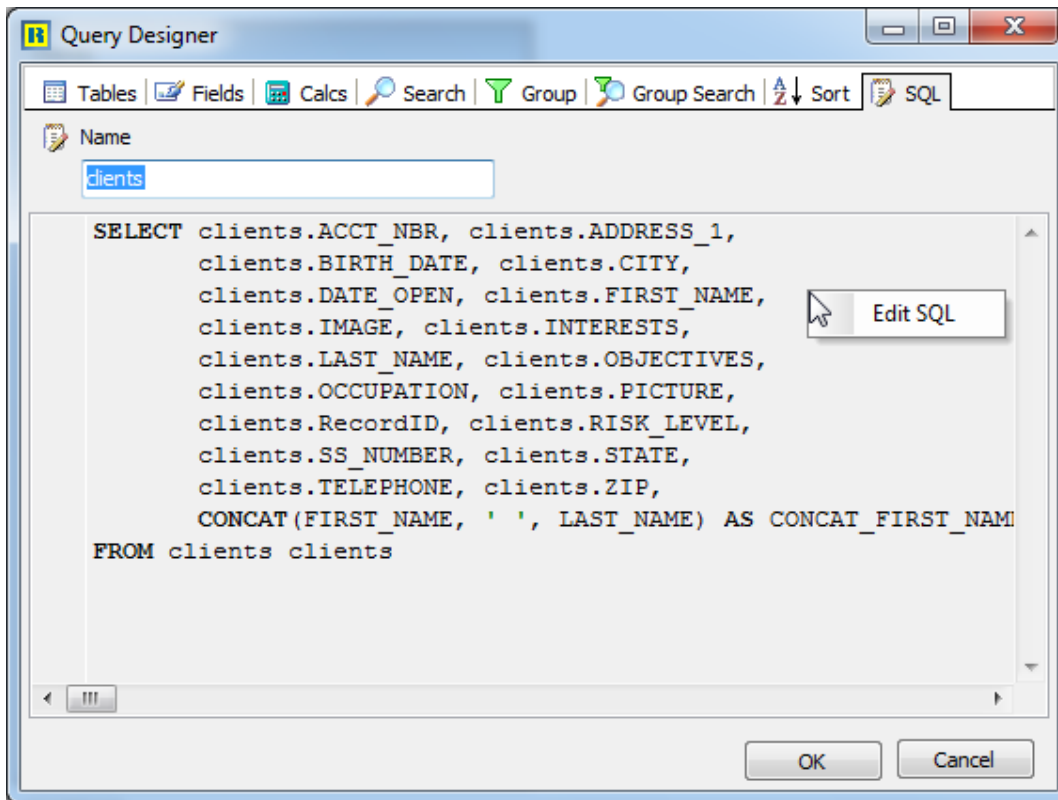
**Note:** Once you manually edit the SQL, you can no longer use the visual design tabs to modify the query.

Perform these steps in order to edit the SQL generated by the Query Designer:

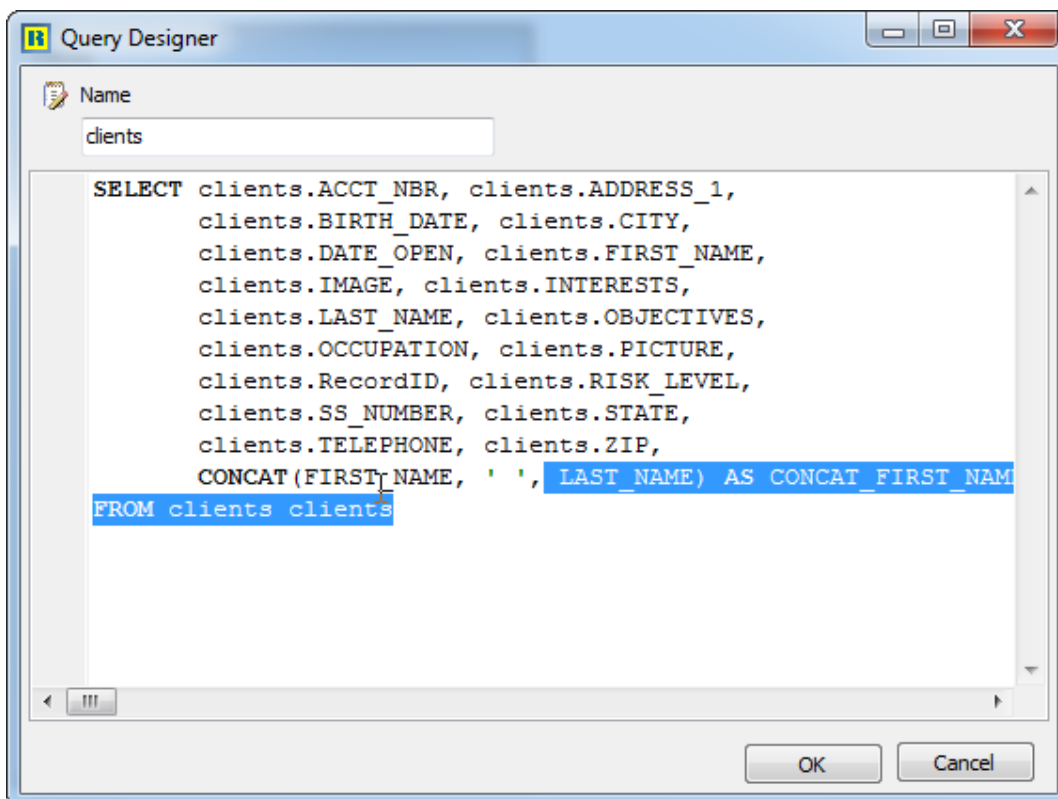
1. **Click on the SQL icon to launch the Query Designer.**



2. Right-click over the SQL text to display the popup-menu.



3. Click Yes to the message dialog. You can now edit the SQL.

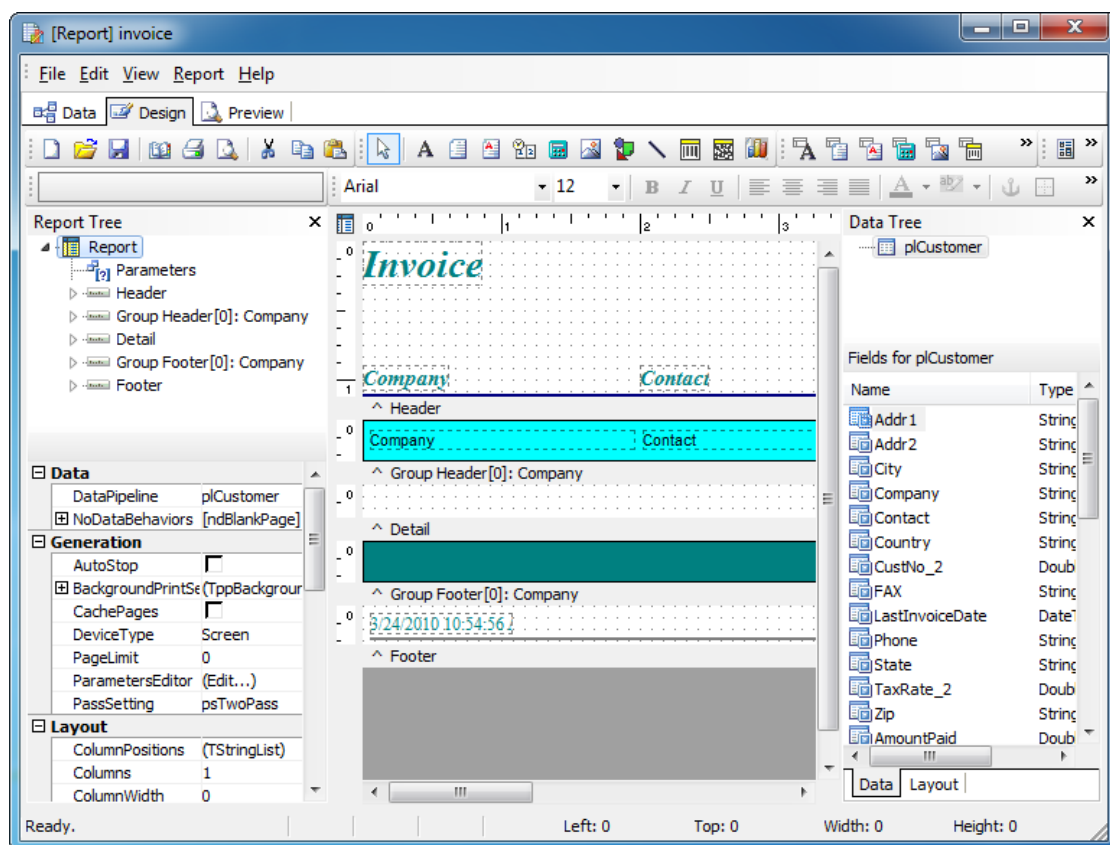




## Report Design

The **Design** tab to build and design reports. The design workspace is divided into two areas: the workbench and the canvas. The workbench is comprised of toolbars, component palettes, rulers, and other tools that can be used to manipulate the canvas. The canvas is the area that contains the report layout. This is where we place the bands and components that will ultimately control the content of each page of the report.

- [Report Wizard](#)
- [Report Design Toolbars](#)
- [Dialogs](#)
- [Report Layouts](#)

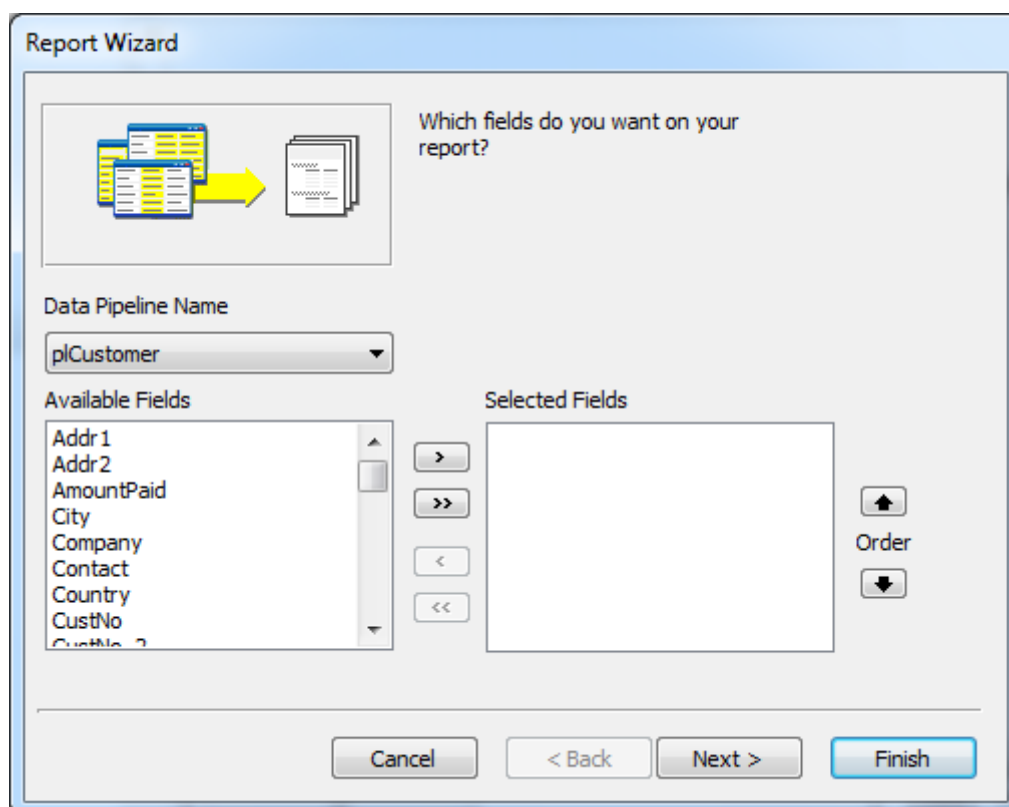


## Report Wizard

The **Report Wizard** is one of the many parts of ReportBuilder that reflects a level of professionalism and attention to detail found in no other reporting product. If you or your end users have utilized standard Windows wizards in other products, then you will be able to quickly recognize and use the ReportBuilder Report Wizard.

The Report Wizard can be accessed via the **File -> New...** menu option of the Report Design tab. A series of screens are presented, each requesting information about the report. When the last page is reached, either a preview or design option can be selected. When the Finish button is clicked, it causes a report to be created and displayed as requested.

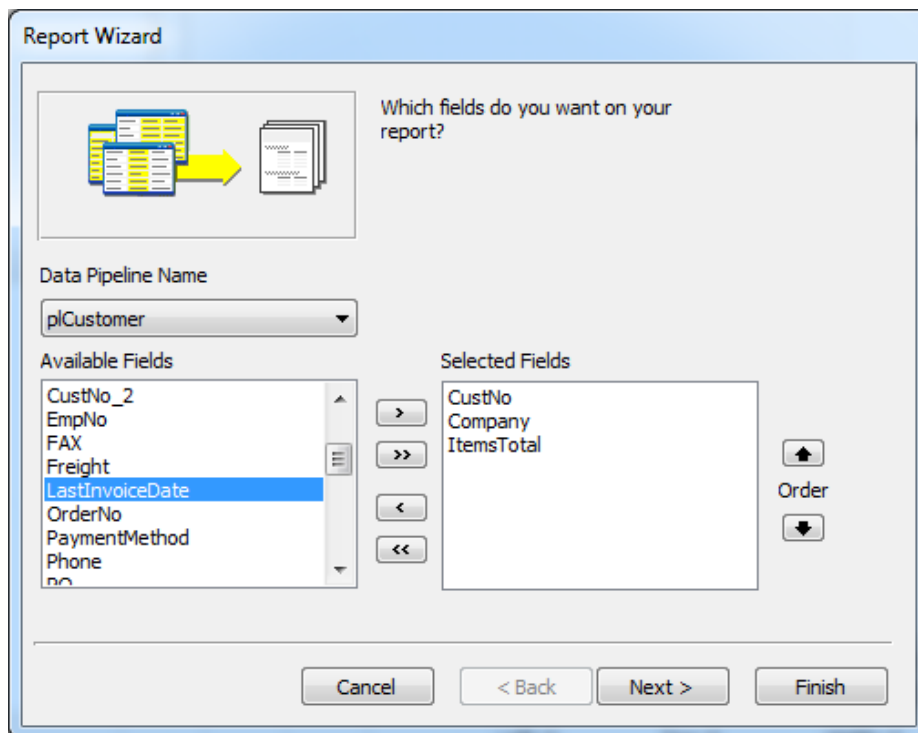
- [Create a Simple Report](#)
- [Create a Group-Based Report](#)



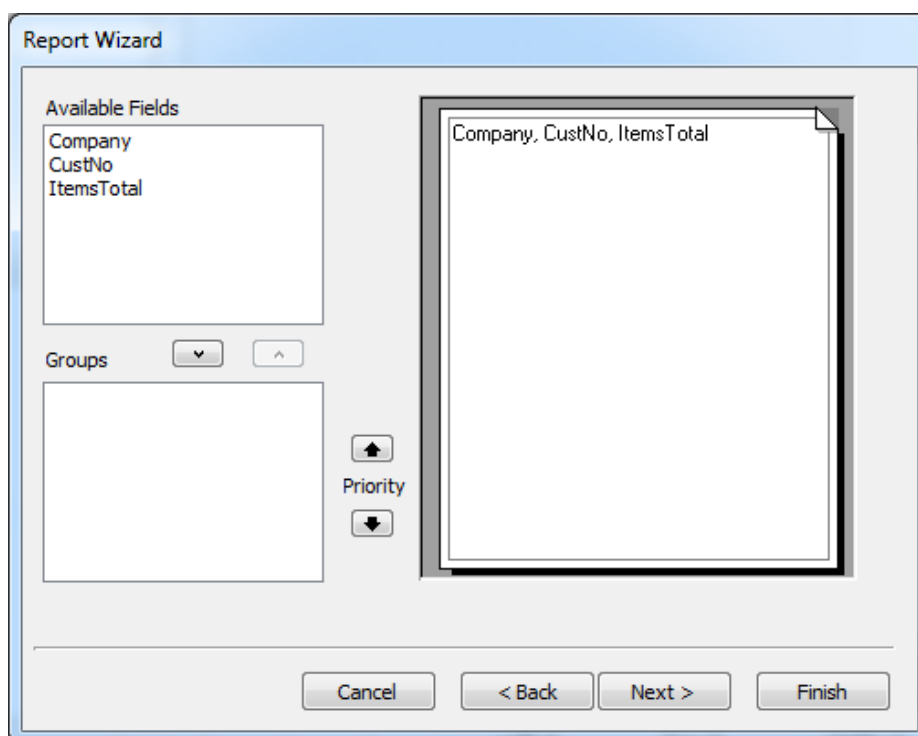
## Create a Simple Report

The following screenshots step through the creation of a simple report via the Report Wizard.

### 1. Select the fields.



### 2. Skip the groups page.



### 3. Select the layout.

Report Wizard

How would you like to lay out your report?

Customers

COMPANY	REGION	CITY	STATE
Action Club	Southwest	Sarasota	FL
Action Club	Southwest	Tampa	FL
Action Club	Southwest	San Jose	CA
Action Club	Southwest	Phoenix	AZ
Action Club	South	Dallas	TX
Action Club	South	Atlanta	GA
Action Club	South	El Paso	TX
Action Diver	Southwest	Charlotte	NC
Action Diver	Southwest	Miami	FL
Action Diver	Southwest	Columbia	SC
Action Diver	Southwest	Lexington	VA

Layout


☐ Vertical

☒ Tabular

Orientation

☒ Portrait

☐ Landscape



☒ Adjust field widths so all fields fit on page.

Cancel < Back Next > Finish

### 4. Select the style.

Report Wizard

What style would you like?

Customers

COMPANY	REGION	CITY	STATE
Action Club	Southwest	Sarasota	FL
Action Club	Southwest	Tampa	FL
		CA	
		AZ	
		TX	
		GA	
		TX	
		NC	
		FL	
		SC	
		VA	

Company  
Action Club

Bold

Casual

Compact

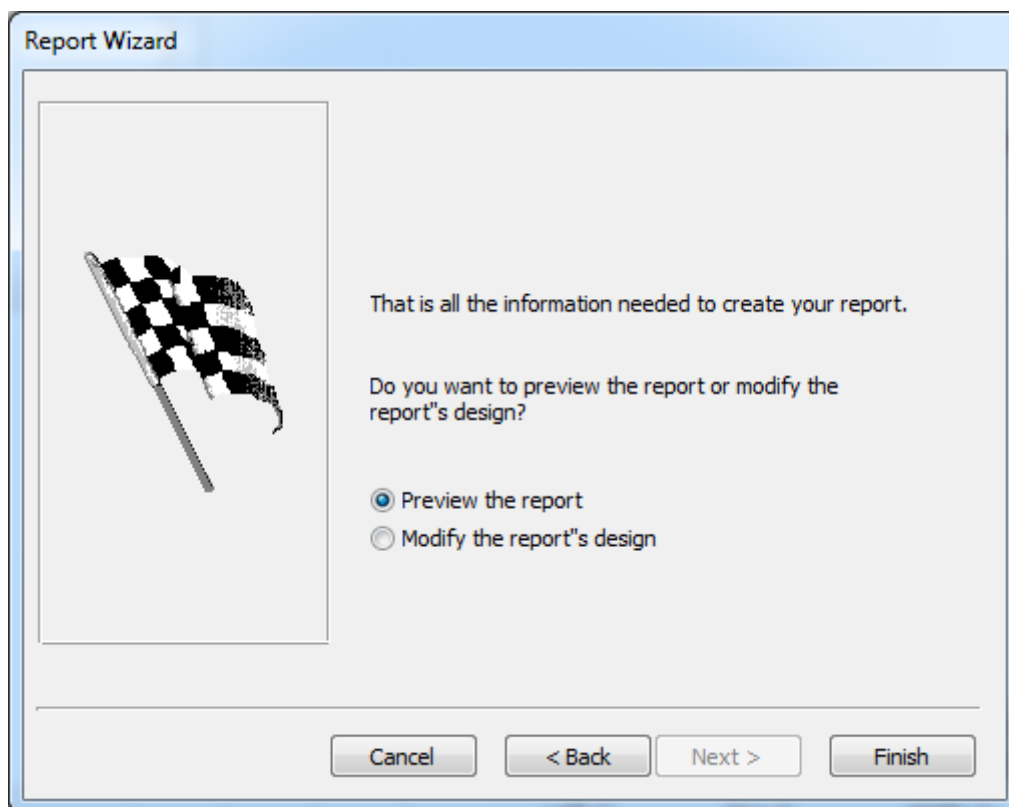
Corporate

Formal

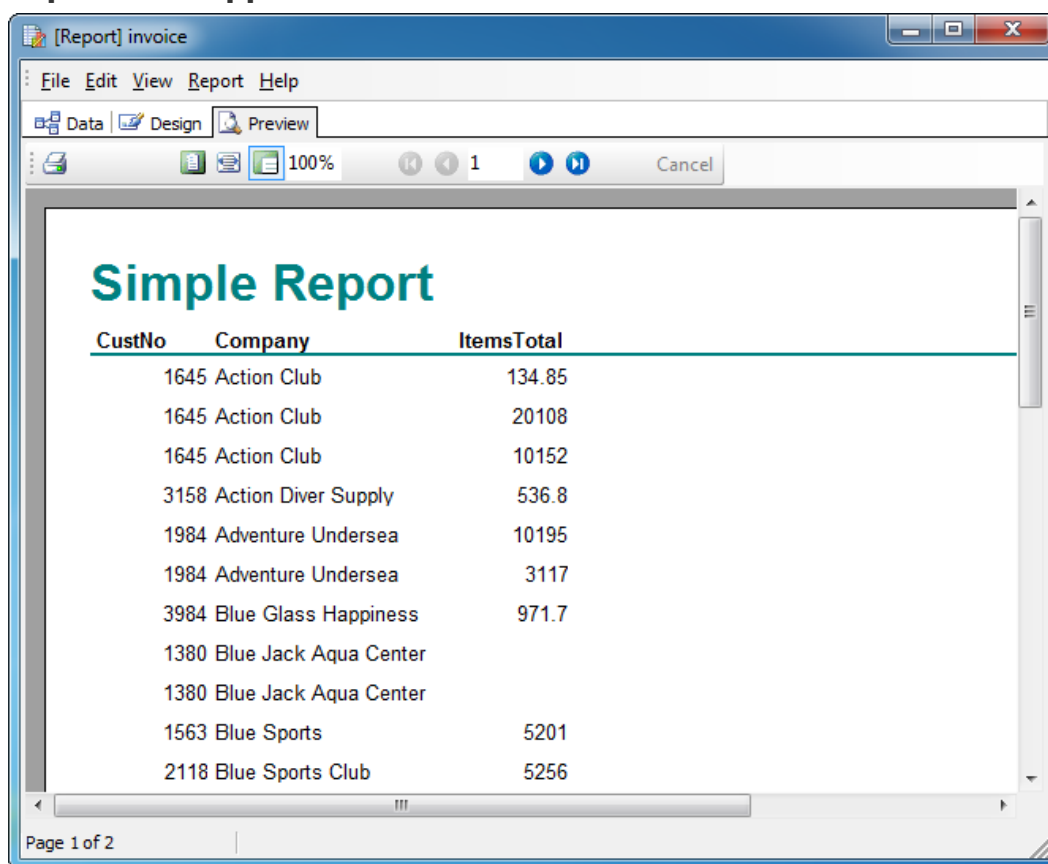
Soft Gray

Cancel < Back Next > Finish

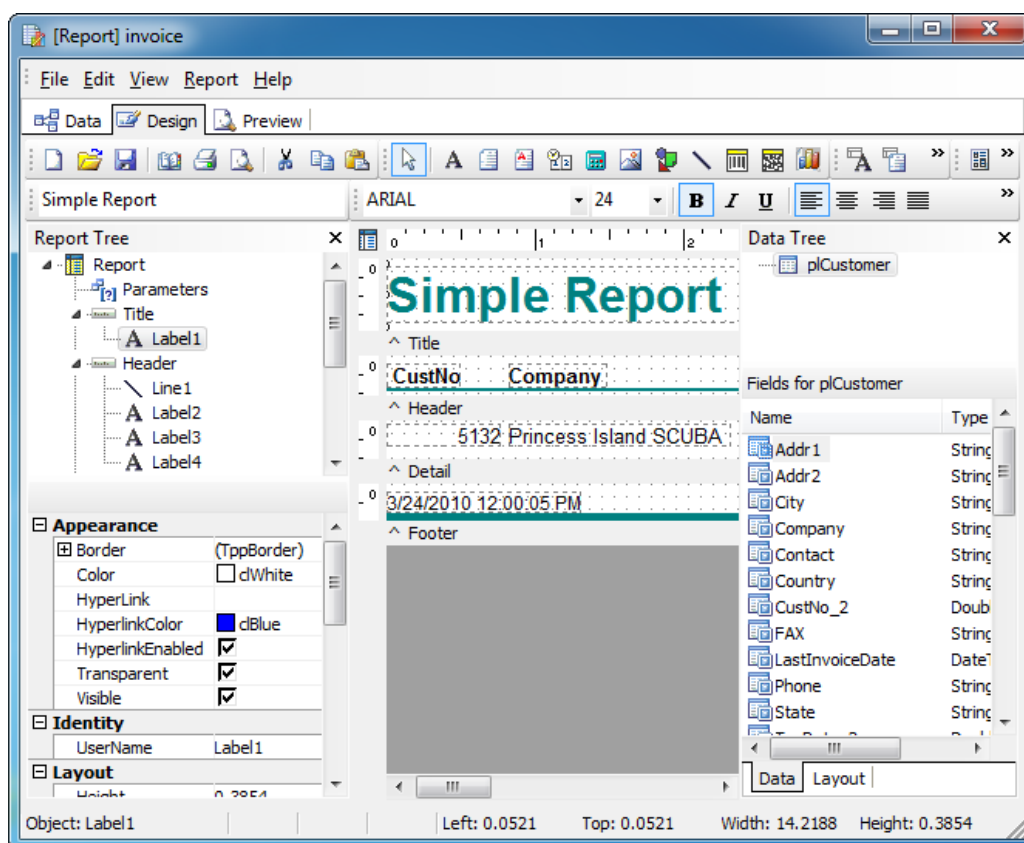
## 5. Select Design or Preview and Finish.



## 6. Report as it appears in the Preview tab.



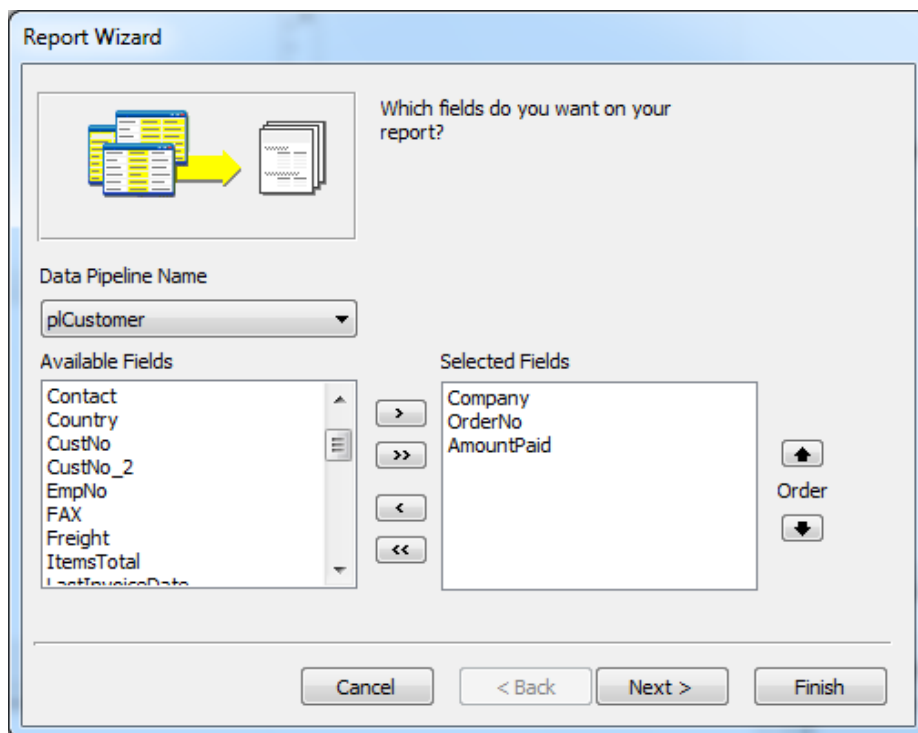
## 7. Report as it appears in the Design tab.



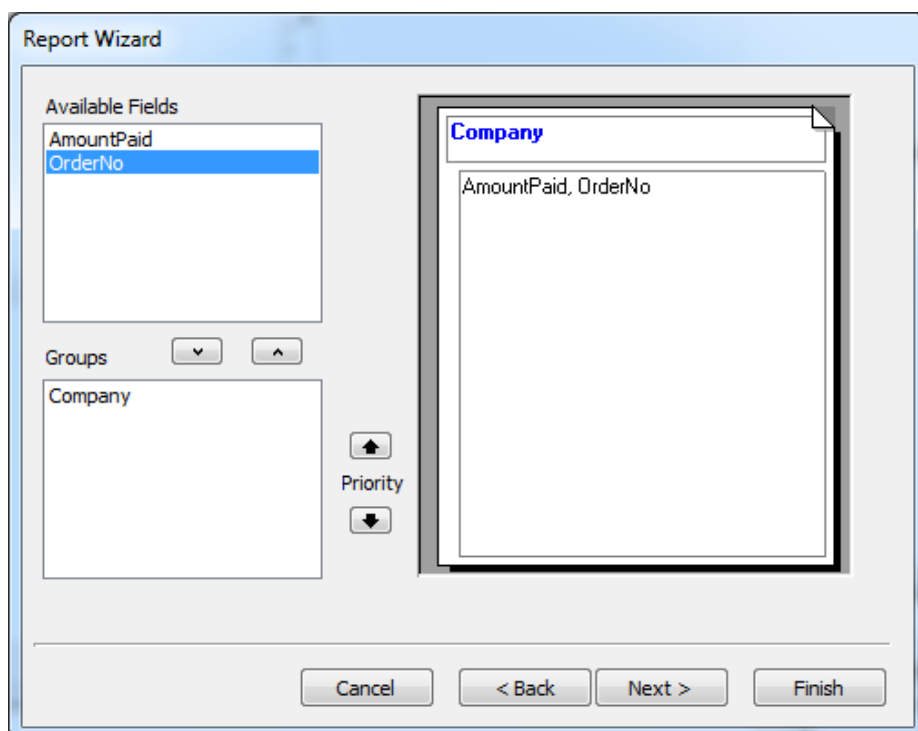
## Create a Group-Based Report

The following screenshots step through the creation of a group-based report via the Report Wizard.

### 1. Select the fields.



### 2. Select a group field.



### 3. Select the layout.

Report Wizard

Orientation

Customers

COMPANY	REGION	CITY	STATE
Action Club	Southwest	Sarasota	FL
		Tampa	FL
	Southwest	San Jose	CA
		Phoenix	AZ
	South	Dallas	TX
		Atlanta	GA
		El Paso	TX
Action Diver	Southwest	Charlotte	NC
		Miami	FL
		Columbia	SC
		Lexington	VA

Layout

☐ Stepped  
☒ Block  
☐ Outline 1  
☐ Outline 2  
☐ Align Left 1  
☐ Align Left 2

Orientation

☒ Portrait  
☐ Landscape

☒ Adjust field widths so all fields fit on page.

Cancel < Back Next > Finish

### 4. Select the style.

Report Wizard

What style would you like?

Customers

COMPANY	REGION	CITY	STATE
Action Club	Southwest	Sarasota	FL
Action Club	Southwest	Tampa	FL
		San Jose	CA
		Phoenix	AZ
		Dallas	TX
		Atlanta	GA
		El Paso	TX
		Charlotte	NC
		Miami	FL
		Columbia	SC
		Lexington	VA

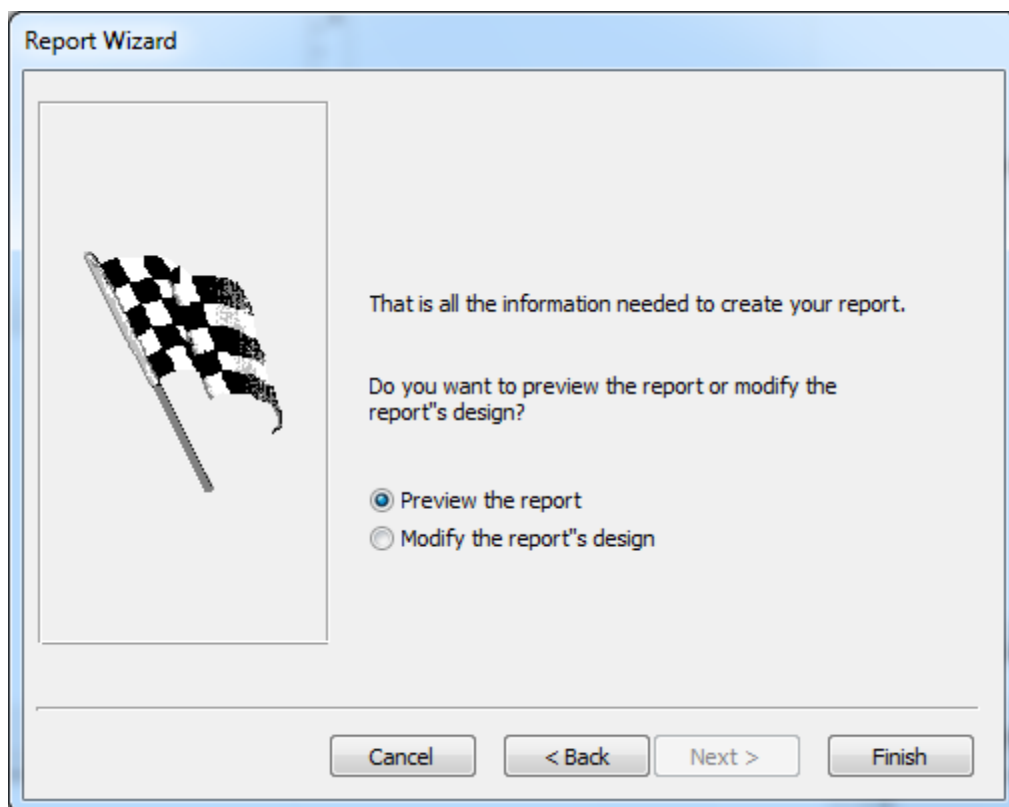
Company  
Action Club

Bold  
 Casual  
 Compact  
 Corporate  
 Formal  
 Soft Gray

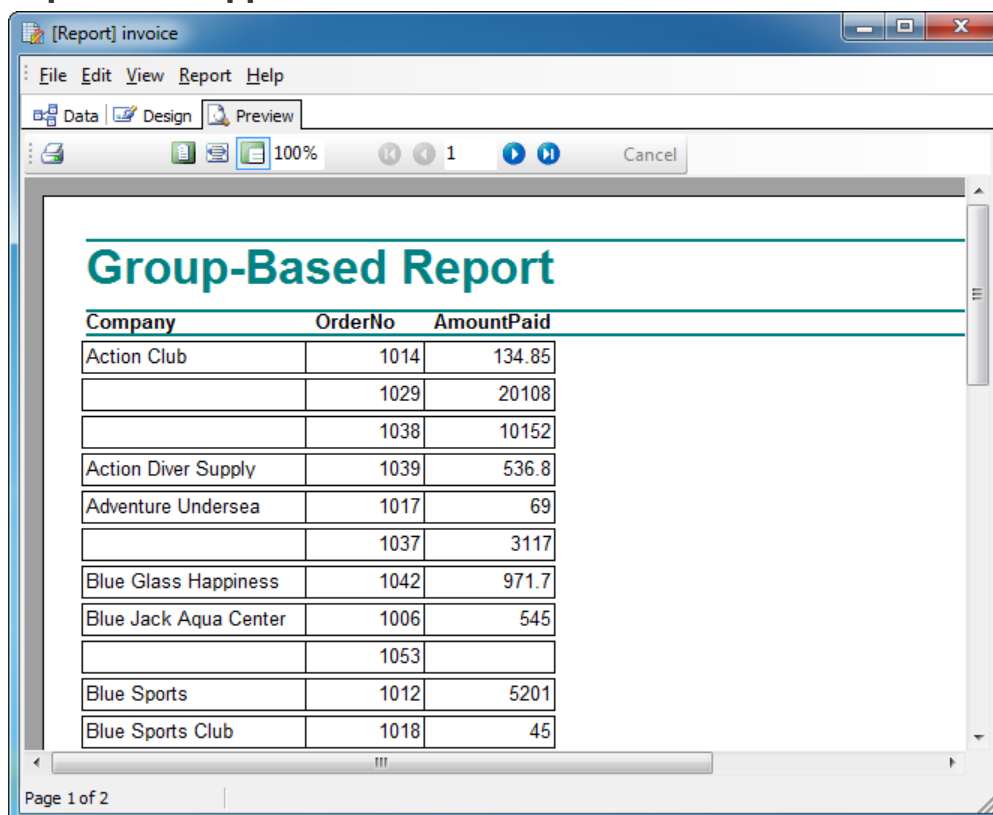
Cancel < Back Next > Finish



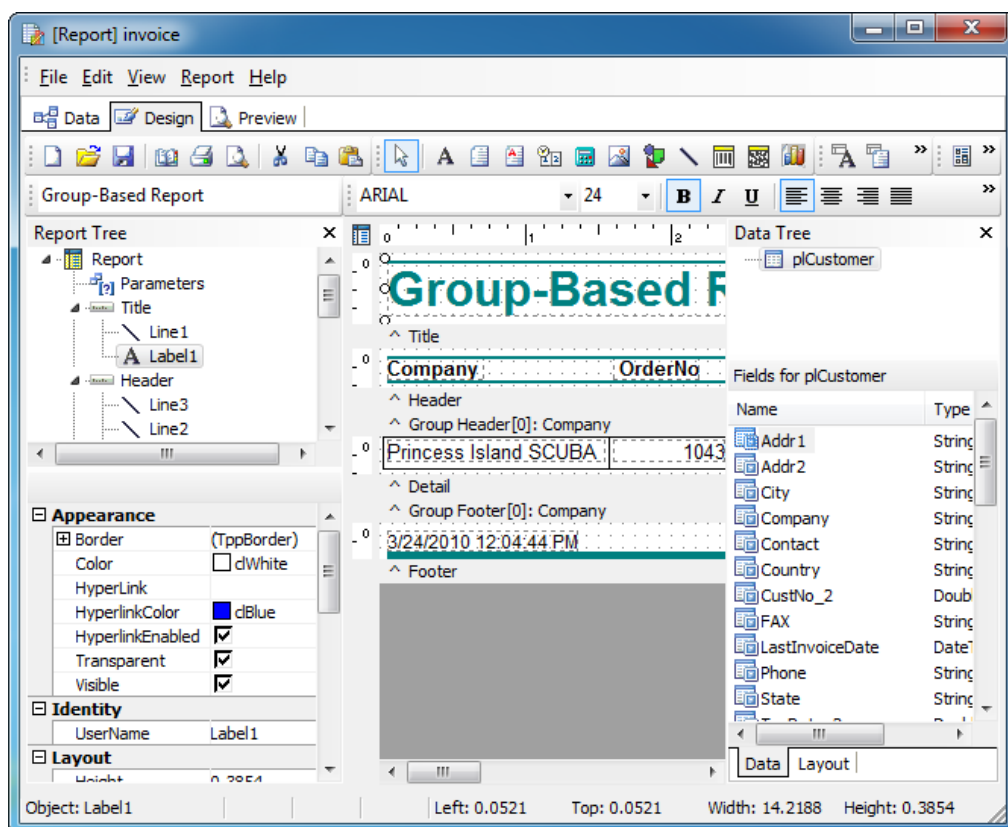
## 5. Select Design or Preview and Finish.



## 6. Report as it appears in the Preview tab.



## 7. Report as it appears in the Design tab.



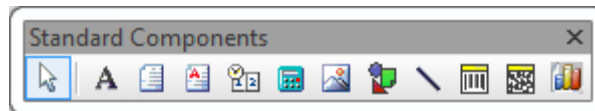
## Report Design Toolbars

The various toolbars accessible from the design workspace are documented in this section. The toolbars are dockable. The toolbars are accessible from the **View -> Toolbars** menu option of the Report Designer or by right-clicking on the docking area at the top of the Report Designer.

- [Standard Component Palette](#)
- [Data Component Palette](#)
- [Advanced Component Palette](#)
- [Report Tree](#)
- [Data Tree](#)
- [Align or Space Toolbar](#)
- [Size Toolbar](#)
- [Nudge Toolbar](#)
- [Draw Toolbar](#)
- [Edit Toolbar](#)
- [Standard Toolbar](#)
- [Format Toolbar](#)

## Standard Component Palette

To access this toolbar, select the **View -> Toolbars -> Standard Components** menu option from the Report Designer main menu. This toolbar will assist in creating the most commonly used report components.



### **Label**

Used to display text. Assign the Caption property to control the text value. To resize the label automatically so it fits a changing caption, set the Auto-Size property to True.

### **Memo**

Prints multiple lines of plain text in a report. To set the value, assign a string list to the Lines property. To dynamically resize the memo during printing, set the Stretch property to True. Use the ShiftRelativeTo property to define dynamic relationships with other stretchable objects.

### **RichText**

Prints formatted text. To set the value, assign the RichText property or use the LoadFromFile or LoadFromRTFStream methods. Use the ShiftRelativeTo property to define dynamic relationships with other stretchable objects. At design-time you can use the ReportBuilder's built-in RTF Editor to load, modify, and save rich text data stored in files.

### **SystemVariable**

Used to display common report information such as page number, page count, print date and time, date, time, etc. The type of information displayed is controlled by the VarType property. The format is controlled by the DisplayFormat property.

### **Variable** (Navicat Report does not support this control)

Used for calculations via an Object Pascal event handler assigned to the OnCalc event or a RAP event handler assigned to the OnCalc event. Access the Calculations dialog (via the speed menu) or the Calc tab of the Report Designer to code a RAP calculation for this component.



## **Image**

Used to display bitmaps and windows metafiles in reports. Assign the Picture property of this component in order to place an image in your report. Use the Report Designer's built-in picture dialog to load images at design-time.

## **Shape**

Use this component to print various shapes (squares, rectangles, circles, ellipses). Set the Shape property to select a type of shape. Use the Brush and Pen properties to control the color and border respectively.

## **Line**

Displays single and double lines (either vertical or horizontal.) Set the Style property to control whether the line is single or double. Set the Weight property to control the line thickness in points. Set the Position property to control whether the line is vertical or horizontal.

## **BarCode**

Used to render barcodes. The string value assigned to the Data property is encoded based on the Bar-CodeType. If the data to be encoded is in a database, use DBBarCode. The following symbologies are supported: Codabar, Code 128, Code 39, EAN-13, EAN-8, FIM A,B,C, Interleaved 2 of 5, Post-Net, UPC-A, UPC-E.

## **2DBarCode**

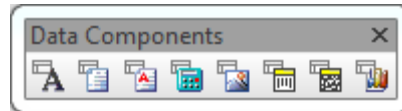
Used to render two-dimensional barcode symbologies. Supports PDF417 and MaxiCode barcode types.

## **Chart**

Used to display standard (non-data-aware) Charts. This component enables you to use Chart inside the Report Designer. You can access the Chart editor via a popup menu.

## Data Component Palette

To access this toolbar, select the **View -> Toolbars -> Data Components** menu option from the Report Designer main menu. This toolbar will assist in creating data-aware report components.



### **ADBText**

Used for displaying values from all types of database fields. Use the DisplayFormat property to format the value.

### **DBMemo**

Used to print plain text from a memo field of a database table. This control will automatically word-wrap the text. Set the Stretch property to True and the component will dynamically resize to print all of the text. Use the ShiftRelativeTo property to define dynamic relationships with other stretchable objects.

### **DBRichText**

Used to print formatted text from a memo or BLOB field of a database table. This control will automatically word-wrap the text. Set the Stretch property to True and the component will dynamically resize to print all of the text. Use the ShiftRelativeTo property to define dynamic relationships with other stretchable objects.

### **DBCalc**

Used for simple database calculations (Sum, Min, Max, Count and Average). The value can be reset when a group breaks using the ResetGroup property.

### **DBImage**

Used to print bitmaps or windows metafiles, which are stored in a database BLOB field.

### **DBBarcode**

Used to render barcodes based on the BarCode-Type and the value supplied via the DataField property. The following symbologies are supported: Codabar, Code 128, Code 39, EAN-13, EAN-8, FIM A,B,C, Interleaved 2 of 5, PostNet, UPC-A, UPC-E.

## **DB2DBarcode**

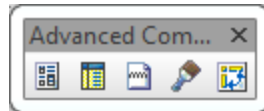
Used to render two-dimensional barcode based on the BarCode Type and the value supplied via the DataField property. The following symbologies are supported: PDF417, MaxiCode.

## **DBChart**

Allows data-aware Charts to be placed within a report.

## Advanced Component Palette

To access this toolbar, select the **View -> Toolbars -> Advanced Components** menu option from the Report Designer main menu. This toolbar will assist in creating advanced report components.



### **Region**

Used to logically group components together. Use the ShiftRelativeTo property to move the region in relation to another dynamically resizing component (such as Memo, RichText, or child-type Sub-Report).

### **SubReport**

Used to handle multiple master details, create sideby-side reporting effects, and hook reports together as one. If you need a report to print within the context of a band, use a child-type subreport. If you need to hook reports together, use a section type subreport. The PrintBehavior property determines the subreport type.

### **PageBreak**

The TppPageBreak component is a report control that allows the user to force a new page during report generation. Placing a TppPageBreak component on a report will cause all objects created after the PageBreak (Z-Order) to be moved to the next page in the report relative to the PageBreak object's top position.

### **PaintBox**

Create a paintbox area for drawing.

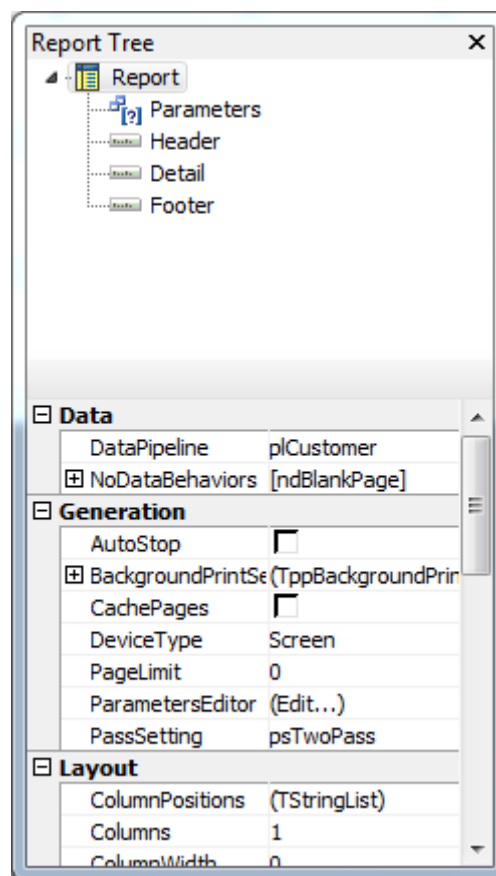
### **CrossTab**

Used to present summarized data in a grid format.



## Report Tree

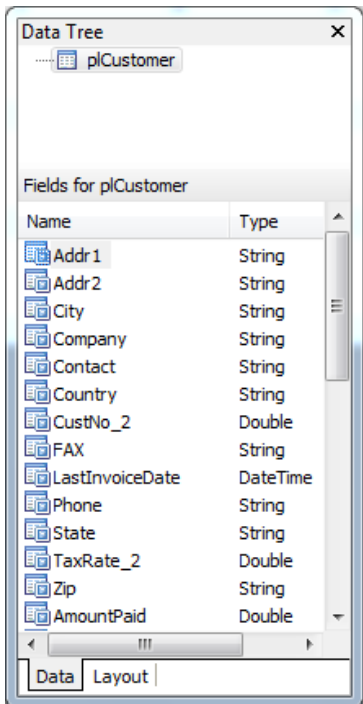
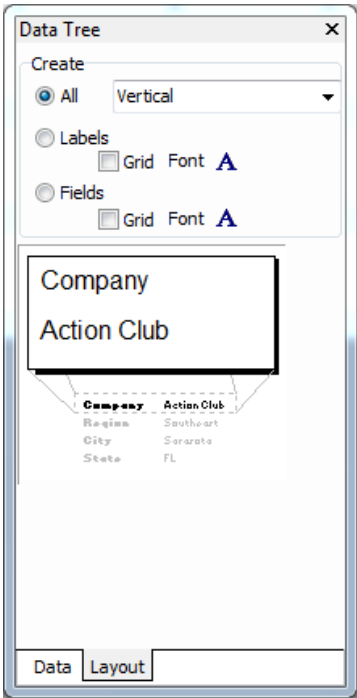
To access this tool window, select the **View -> Toolbars -> Report Tree** menu option from the Report Designer main menu. This tool window is dockable on the left and right sides of the Report Designer. It can be used to organize components within each band. Components selected in the Report Tree are selected in the report layout. The upper portion of the Report Tree shows the main report object and any subreports nested within it. This feature can be helpful for organizing your subreports.



## Data Tree

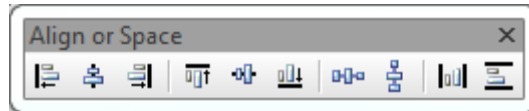
To access this tool window, select the View -> Toolbars -> Data Tree menu option from the Report Designer main menu. This tool window is dockable on the left and right sides of the Report Designer. It can be used to create components within any band. Simply select a set of fields and drag the selection into the band. A set of corresponding data-aware components will be created.

The Data Tree has two tabs:

	
<p><b>Data Tab</b></p> <p>In the top tree view, the Data tab contains a list of data pipelines to which the report has access. In the bottom list view, all of the fields for the currently selected data pipeline are displayed. Fields can be selected from the bottom list view and dragged to any part of the report layout. The data-aware component that is appropriate for the given field will then be created along with a label and border.</p>	<p><b>Layout Tab</b></p> <p>This tab allows you to control drag-and-drop behavior. A label and border are created for each data-aware component by default. You can turn the label and the border off, control the color of the label or border, and control the font of the label and data-aware component from this tab. Once you've set the drag-and-drop behavior, it will be retained for future design sessions.</p>

## Align or Space Toolbar

To access this toolbar, select the **View -> Toolbars -> Align or Space** menu option from the Report Designer main menu. This toolbar will assist in positioning components relative to one another and relative to the band in which they appear.



### **Align Left**

Aligns a group of components with the leftmost position of the component that was selected first.

### **Align Middle**

Centers a group of components based on the horizontal center of the component that was first selected.

### **Align Right**

Aligns a group of components with the rightmost position of the component that was selected first.

### **Align Top**

Aligns a group of components with the topmost position of the component that was selected first.

### **Align Center**

Aligns a group of components based on the vertical center of the component that was first selected.

### **Align Bottom**

Aligns a group of components with the bottommost position of the component that was selected first.

### **Space Horizontally**

Spaces a set of components based on the leftmost position of the first component selected and the rightmost position of the last component selected.

## **Space Vertically**

Spaces a set of components based on the topmost position of the first component selected and the bottommost position of the last component selected.

## **Center Horizontally in Band**

Centers a component horizontally within a band.

## **Center Vertically in Band**

Centers a component vertically within a band.

## Size Toolbar

To access this toolbar, select the **View -> Toolbars -> Size** menu option from the Report Designer main menu.



### **Shrink Width to Smallest**

Determines the minimum width of all the selected components, and then sets the width of the components to that value.

### **Grow Width to Largest**

Determines the maximum width of all the selected components, and then sets the width of the components to that value.

### **Shrink Height to Smallest**

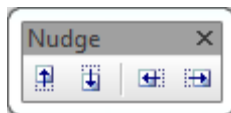
Determines the minimum height of all the selected components, and then sets the height of the components to that value.

### **Grow Height to Largest**

Determines the maximum height of all the selected components, and then sets the height of the components to that value.

## Nudge Toolbar

To access this toolbar, select the **View -> Toolbars -> Nudge** menu option from the Report Designer main menu.



### **Nudge Up**

Moves all selected components one pixel up.

### **Nudge Down**

Moves all selected components one pixel down.

### **Nudge Left**

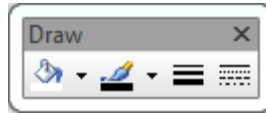
Moves all selected components one pixel to the left.

### **Nudge Right**

Moves all selected components one pixel to the right.

## Draw Toolbar

To access this toolbar, select the **View -> Toolbars -> Draw** menu option from the Report Designer main menu. This toolbar will assist in setting the colors and borders of components.



### **Fill Color**

For shapes, lines, and region components only. Sets the Brush.Color property. To set the color of a textual component, check the Highlight Color action of the Format toolbar.

### **Line Color**

For shapes, lines, and region components only. Sets the Pen.Color property.

### **Line Thickness**

For use with a Line component only. Sets the Weight property.

### **Line Style**

For use with a Line component only. Sets the Pen.Style property.

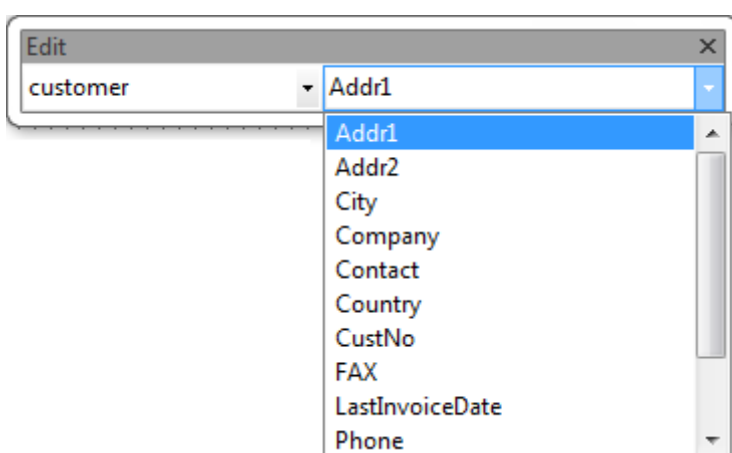
## Edit Toolbar

To access this toolbar, select the **View -> Toolbars -> Edit** menu option from the Report Designer main menu. This toolbar will assist in setting the most important property or properties for the currently selected component.

### 1. No component selected

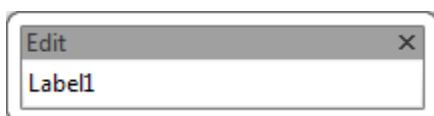


### 2. Data-aware component selected



This configuration allows the data pipeline and data field for the component to be set. The drop-down list on the left shows the data pipeline. The drop-down list on the right shows the field name.

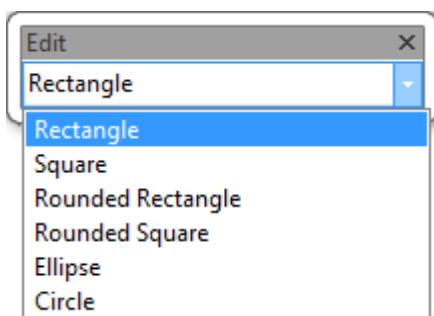
### 3. Label component selected



Here a label component has been selected in the Report Designer. The Edit toolbar displays an edit box from which the label's caption can be set.

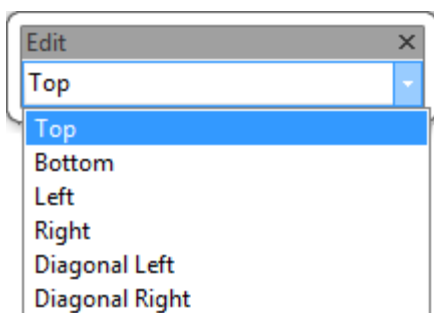


## 4. Shape component selected



Here a shape component has been selected in the Report Designer. The Edit toolbar displays the different shape types.

## 5. Line component selected



This configuration allows you to move the line to the top, bottom, left, or right within the line's selection handles.

## Standard Toolbar

To access this toolbar, select the **View -> Toolbars -> Standard** menu option from the Report Designer main menu. This toolbar will assist with saving the report layout, accessing the print and print preview options, and accessing the cut and paste operations.



### **New**

Creates a blank report layout.

### **Open**

Displays the Open dialog, allowing you to open an existing report layout.

### **Save**

Saves a report layout to file.

### **Page Setup**

Displays the Page Setup dialog, allowing you to set the paper size and configure the layout for the report.

### **Print**

Displays the Print dialog before sending the report to the printer.

### **Print Preview**

Displays the Print Preview window.

### **Cut**

Cuts the currently selected components into the clipboard.

### **Copy**

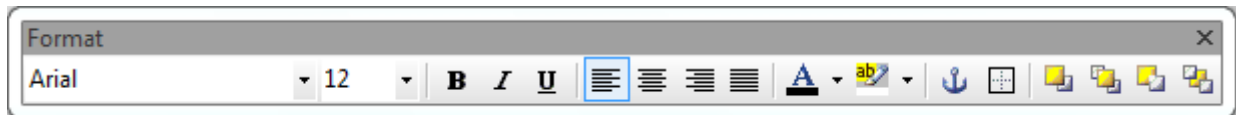
Copies the currently selected components into the clipboard.

### **Paste**

Pastes the components in the clipboard into the report.

## Format Toolbar

To access this toolbar, select the **View -> Toolbars -> Format** menu option from the Report Designer main menu. This toolbar will assist with setting the font and colors. It will also assist with layering the components.



### Font Name

Selects the font name for textual components.

### Font Size

Selects the font size. You can also type in this box to set the font size exactly.

### **B Bold**

Sets the font to bold.

### *I Italic*

Sets font to italic.

### U Underline

Sets font to underline.

### **Left Justify**

Left justifies the text in the component.

### **Center**

Centers the text in the component.

### **Right Justify**

Right justifies the text in the component.

### **Justify**

Justifies the text in the component.

### **Font Color**

Sets the font color.

## **Highlight Color**

Sets the background color of the textual component.

## **Anchors**

Specifies how a report component is anchored to its parent. Use Anchors to ensure that a report element maintains its current position relative to an edge of its parent control (i.e. Band/Region), even if the parent is resized.

## **Border**

Specifies which of the outside border lines of a report component are rendered.

## **Bring to Front**

Brings the component to the front. The components in the front print last, and the components in the back print first. Use the Report Tree to see the exact layering of components within the band.

## **Send to Back**

Sends the component to the back. The components in the front print last, and the components in the back print first. Use the Report Tree to see the exact layering of components within the band.

## Dialogs

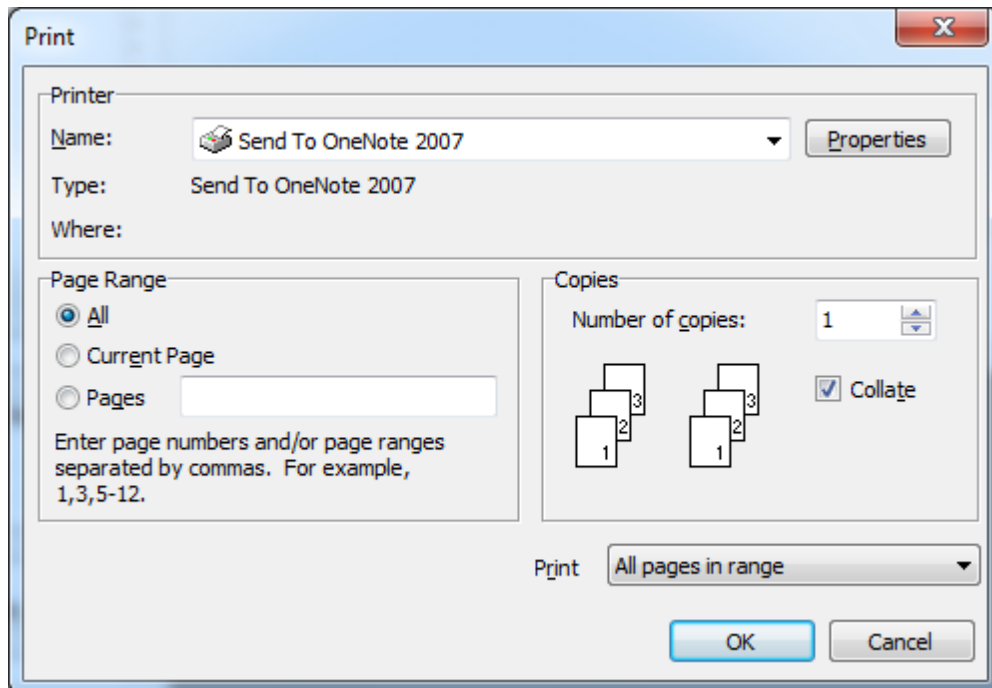
Navicat Report Designer allows you to perform extra tasks: print reports, report page setup, adding groups and outline settings, etc.

- [Print Dialog](#)
- [Page Setup Dialog](#)
- [Groups Dialog](#)
- [Print to File Setup Dialog](#)
- [Data Dialog](#)
- [Grid Options](#)
- [Outline Settings](#)
- [Find Text Settings](#)

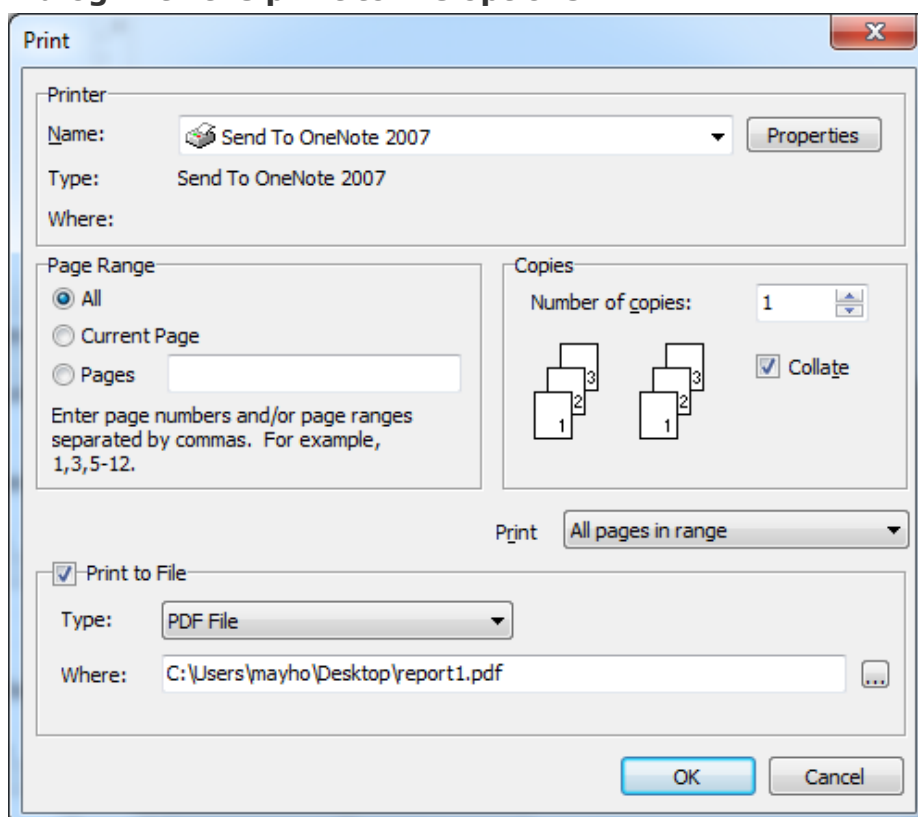
## Print Dialog

The Print Dialog is automatically displayed when the report is sent to the printer, allowing you to select the pages, number of copies, and printer for the report. When the **AllowPrintToFile** or **AllowPrintToArchive** properties of the Report are enabled, this dialog displays additional print to file options.

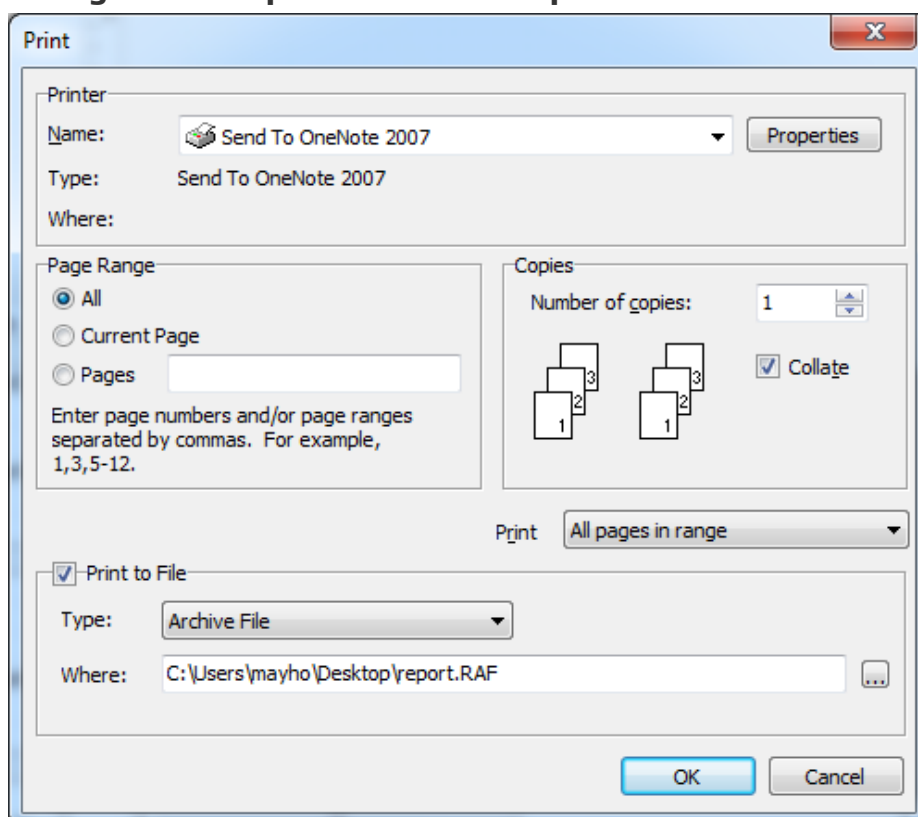
1. **The print job settings can be set via the standard dialog.**



## 2. Dialog with the print to file options.



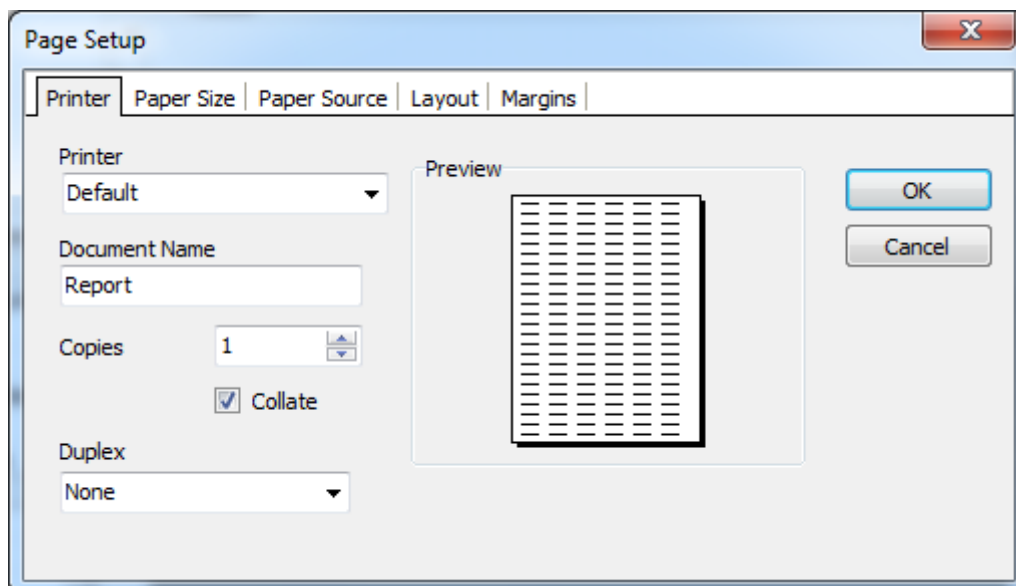
## 3. Dialog with the print to archive options.



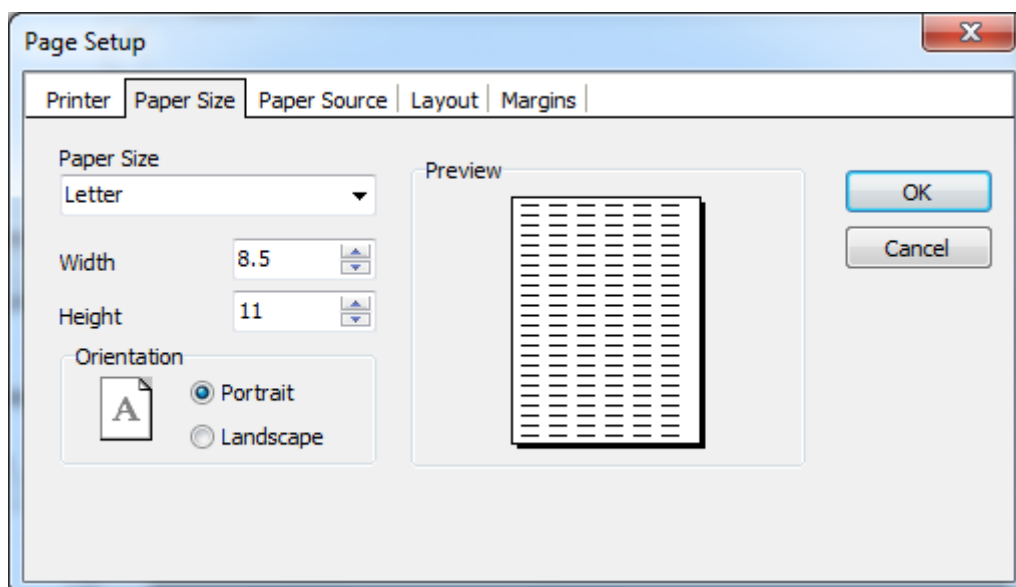
## Page Setup Dialog

The Page Setup dialog can be accessed from the **File -> Page Setup...** menu option of the Report Designer. You can set the following properties from within the Page Setup dialog:

### 1. Printer

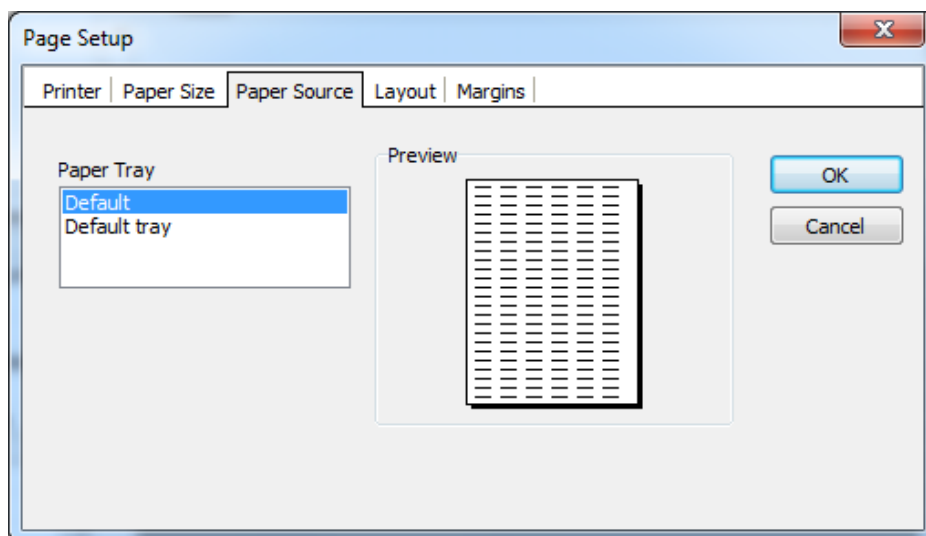


### 2. Paper Size and orientation

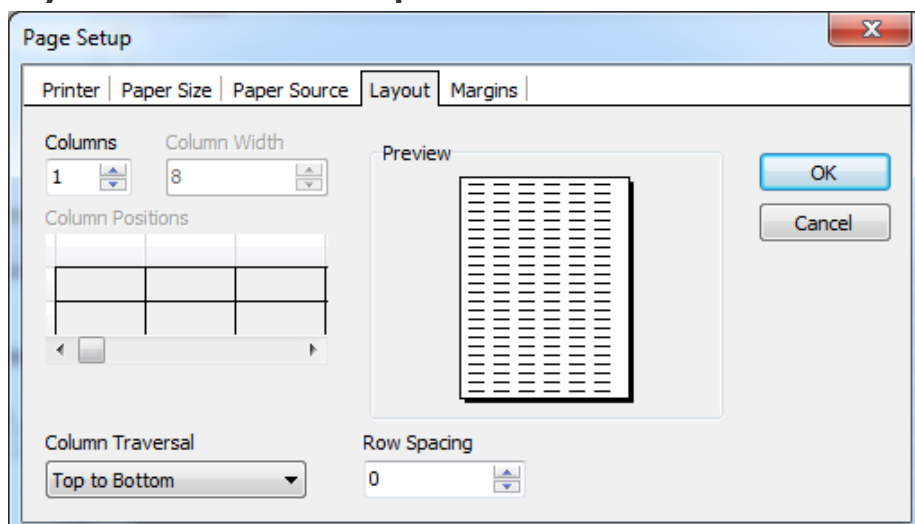




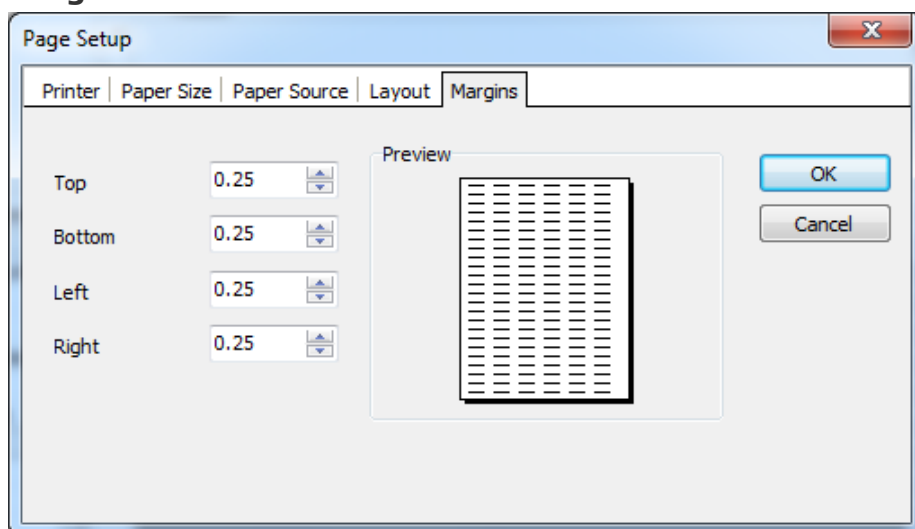
## 3. Paper Source



## 4. Layout for Columnar Reports

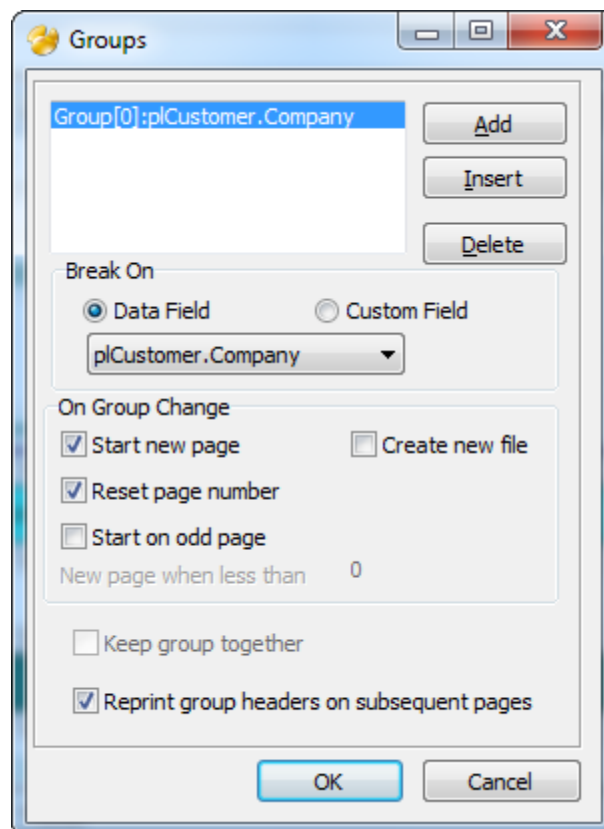


## 5. Margins



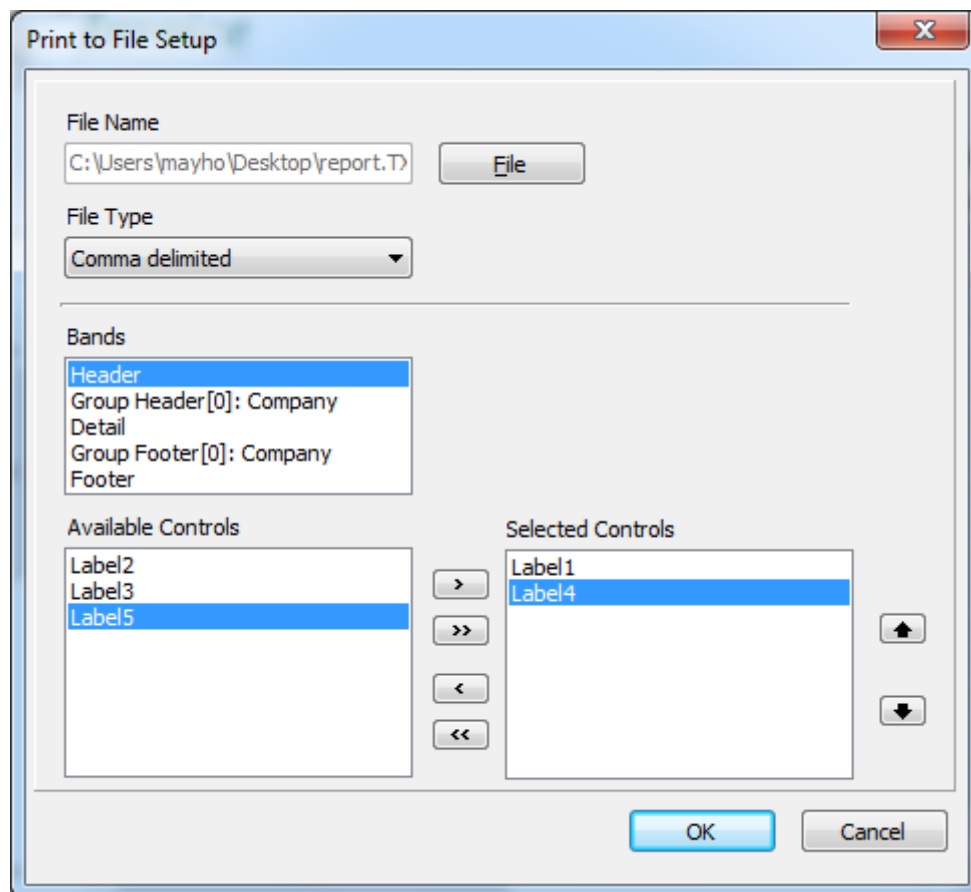
## Groups Dialog

The Groups dialog is accessible via the **Report -> Groups...** menu option of the Report Designer. You can separate your report into different sections using groups. A number of options are available to control the behavior of each group. For example, you may want each group to start on a new page or to reprint the group header when the group continues on additional pages. Another powerful feature is the Keep group together option, which can be used to ensure that all of the information for a group fits on a page.



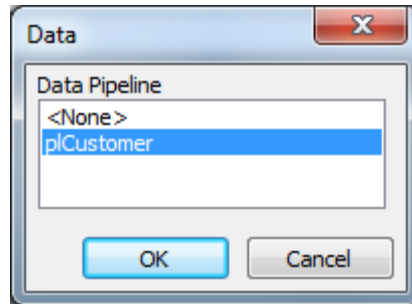
## Print to File Setup Dialog

The Print to File Setup dialog is accessible via the **File -> Print to File Setup...** menu option of the Report Designer. This dialog is used to specify the format and content of the ASCII file that will be created if the report is printed to file.



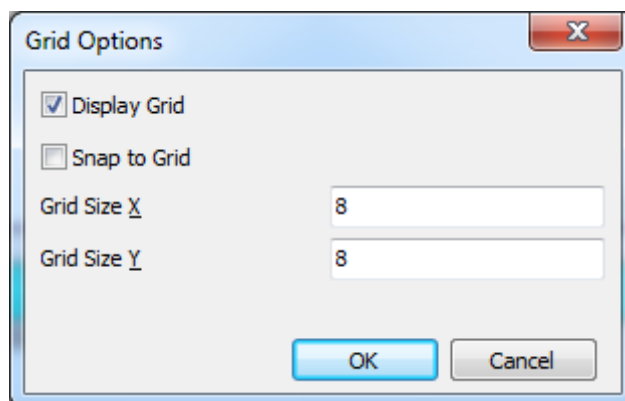
## Data Dialog

The Data dialog can be accessed from the **Report** -> **Data...** menu option of the Report Designer. It can be used to specify the data pipeline for the report.



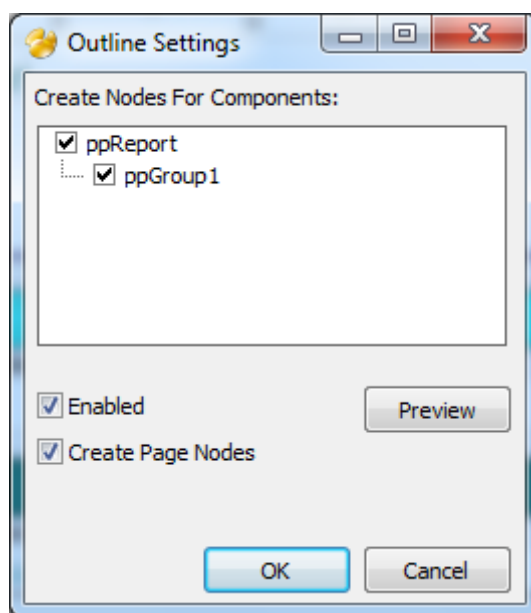
## Grid Options

The Grid Options dialog is accessible via the **View -> Grid Options...** menu of the Report Designer. Use the Grid Options dialog to control how the workspace grid is drawn and whether layout elements automatically snap to the grid points.



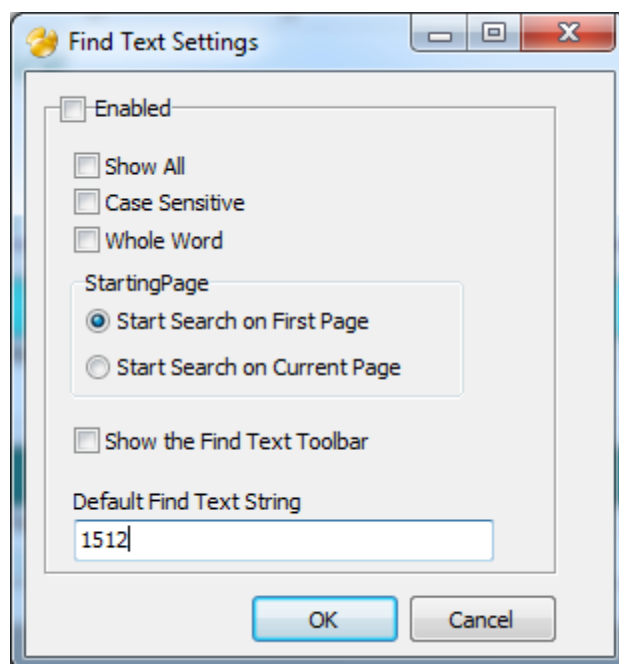
## Outline Settings

The Outline Settings dialog is accessible via the **Report -> Outline Settings...** menu option of the Report Designer. Use this dialog to control the behavior of report outline generation. When enabled, an outline tree structure is dynamically generated by the report engine and rendered by the report previewer.



## Find Text Settings

The Find Text Settings dialog is accessible via the **Report -> Find Text Settings...** menu option of the Report Designer. Use this dialog to configure the find text options used by the report previewer. When enabled, the previewer can be used to find and highlight text that appears in the pages of the report.



## Report Layout

There are many special options to set the layout. You can simply right-click the components and enable the following options:

### **AutoDisplay**

Determine whether to automatically display the contents of a BLOB in a database field in a Memo or Image component.

### **AutoSize**

The report component resizes the width and length.

### **AutoSizeFont**

Set the human readable text to a standard accepted font size based on the size of the barcode component.

### **Child**

The report prints to completion, using the subreport components width as the page width and stretching over several pages until printing is complete.

### **DirectDraw**

Indicate the image will be sent directly to the printer or to an intermediate bitmap and then copied to the printer.

### **Fixed**

A single page prints based on the dimensions and position of the subreport component.

### **MaintainAspectRatio**

Maintain image original width to height ratio.

### **NewPrintJob**

Determine whether a new print job is started when the subreport is sent to the printer.

### **GraphicType**

Indicate the kind of the graphic object: Bitmap, GIF, Icon, JPEG and Metafile.

### **KeepTogether**

Control the pagination of the component when the contents cannot fit on the current page.



## **LookAhead**

Display summary calculations in the title band, page footer calculations in the page header band, group footer calculations in the group header band and column footer calculations in the column header.



## **ParentHeight**

Force the height of the component to match height of the parent component. The parent may be either a band or a region.



## **ParentWidth**

Force the width of the component to match width of the parent component.

## **ParentPrinterSetup**

Determines whether the values for the PrinterSetup property should be copied from the parent report.

## **PrintHumanReadable**

Output the human readable version of the bar code (i.e. the actual letters or numbers).



## **ReprintOnOverflow**

Use to print non-stretching components when stretching components are printing on additional pages.

## **ReprintOnSubsequent**

Use in conjunction with the SuppressRepeatedValues property. When SuppressRepeatedValues is set to True, and detail lines have overflowed onto a new page, you can force the value to reprint on the first detail line of the new page by setting ReprintOnSubsequent to True.

## **ResetPageNo**

Enable subset page numbering. In subset page numbers, the page numbers and page count of the subreport are displayed instead of the master report page numbers.

## **Section**

The report prints as a separate section, starting a new page when it begins and finishing the last page when it ends. The PrinterSetup property is used to determine the page size and printer settings.

## **ShiftRelativeTo**

Use to specify the vertical positioning that should take place between multiple stretching components in a band.

## **Stretch**

Determine whether the image is scaled to fit inside the component.

## **StretchWithParent**

Allow a shape or line to expand or contract based on the change in height of the band or region in which it is contained.

## **SuppressRepeatedValues**

When the value of a field (as determined by the DataField property) is the same across several records, the SuppressRepeatedValues property allows you to print the value only once each time the field value changes.

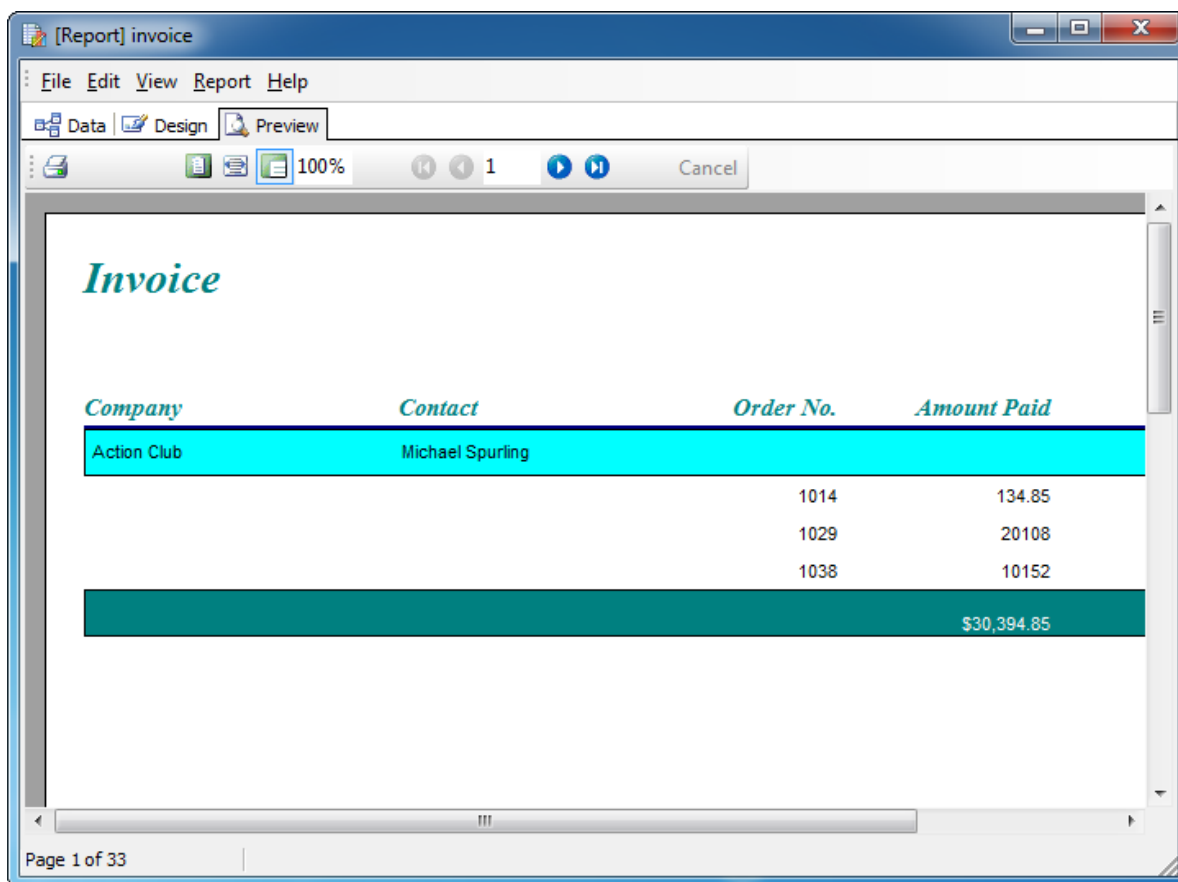
## **TraverseAllData**

Indicate that the subreport will continue to print (print on additional pages in the same exact position) until all data is traversed. By default the fixed type subreport truncates the data.

## Report Preview

The **Preview** tab contains the rendered report. The report can be printed to the printer or to various file formats.

- [Preview and Print](#)
- [Search](#)



The screenshot shows the 'Report Preview' window for an invoice. The window has a menu bar (File, Edit, View, Report, Help) and a toolbar with icons for Data, Design, and Preview. The 'Preview' tab is active, showing a rendered invoice. The invoice has a title 'Invoice' and a table with columns: Company, Contact, Order No., and Amount Paid. The table data is as follows:

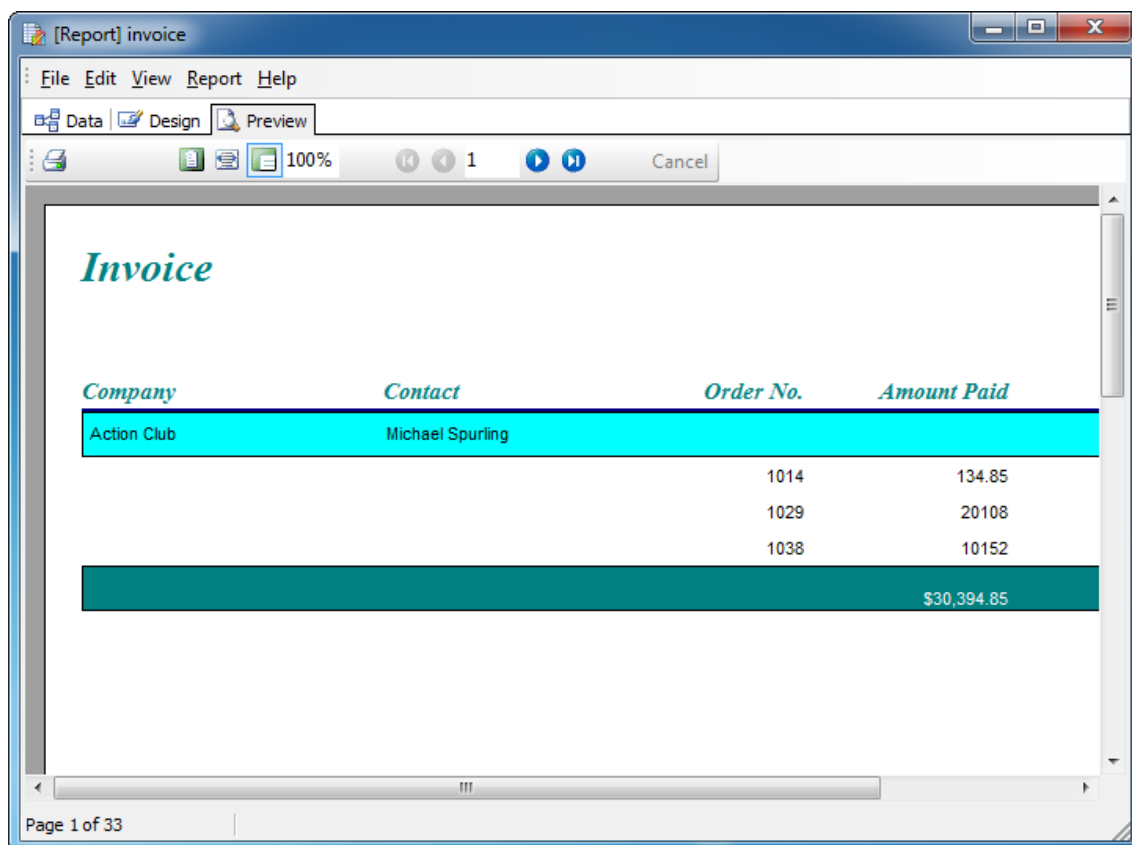
Company	Contact	Order No.	Amount Paid
Action Club	Michael Spurling	1014	134.85
		1029	20108
		1038	10152
			\$30,394.85

The window also shows a status bar at the bottom indicating 'Page 1 of 33'.

## Preview and Print

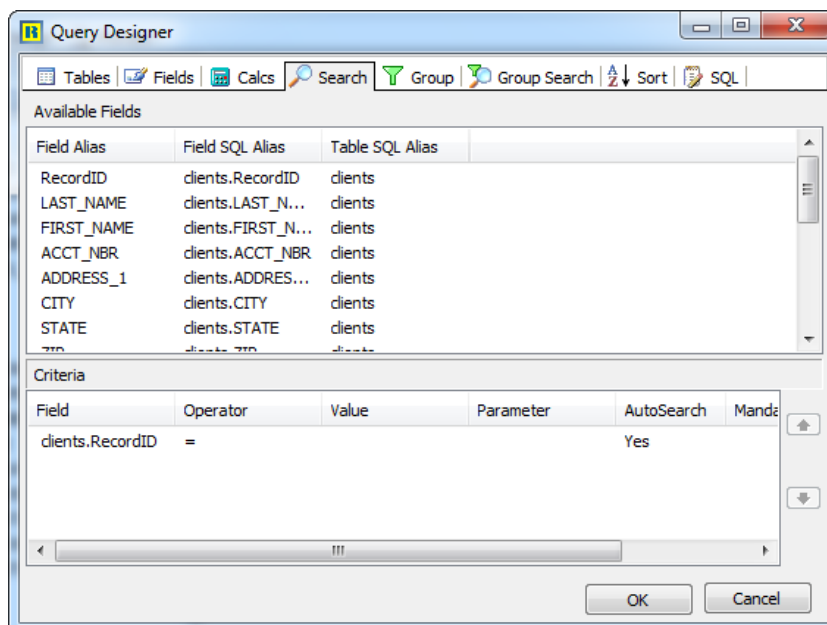
The Preview form enables the user to control and view report output. An outline of page nodes is displayed on the left side of the form, adjacent to the page viewer.

Use the  Print button to send report output to the printer or to an output file, such as PDF.

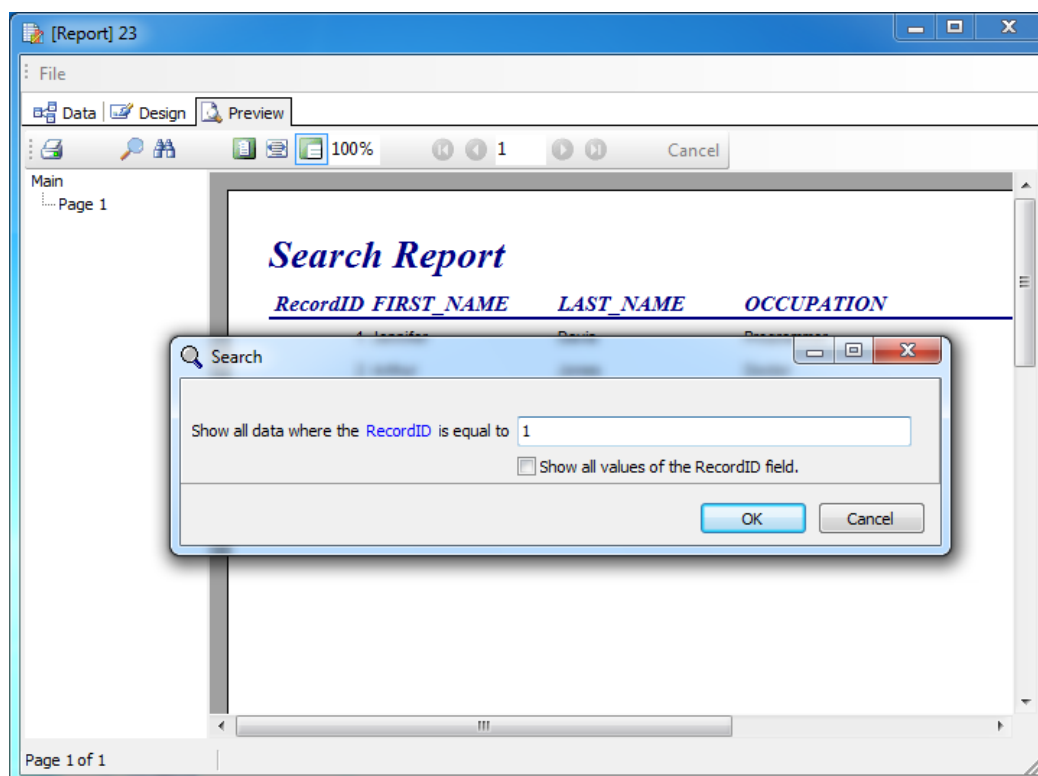


## Search

To create a Search Report, enable the AutoSearch option in the Query Designer.



Then you can click the  **Search Data** icon to enter the searching criteria.



## Tutorial on creating different types of reports (Available only for MySQL)

### How to Start

Before you start the tutorial for reports creation. Please install the Report Samples data on your MySQL server.

#### Steps:

1. Download data file - [http://www.navicat.com/download/report\\_sample/report\\_sample.sql](http://www.navicat.com/download/report_sample/report_sample.sql) and extract it.
2. Create a **new connection** using the name "**report\_sample**"
3. Create a **new database** using the name "**report\_sample**"
4. Choose "Query" icon and click "New Query" button. In Query Window, please click "**Load**" button to load the "**report\_sample.sql**" file.
5. Click "**Run**" button to execute the SQL statements.

### Different Types of Reports

Please click on the links listed as below.

- i. [Simple Report](#)
- ii. [Cover Page](#)
- iii. [Simple Table Listing](#)
- iv. [Invoice Report](#)
- v. [Order Summary Report](#)
- vi. [Detailed Order Summary](#)
- vii. [Sales Statistics](#)
- viii. [Crosstab Report](#)
- ix. [Mailing Label](#)
- x. [Form Letter](#)
- xi. [Photo Album](#)

## Simple Report

### OVERVIEW

This tutorial will solidify all of the report building techniques.

The final report should contain the following items:

- A list of customer numbers
- A list of company names
- A contact for each company
- A phone number for each contact

Custno	Company	Contact	Phone
1221	Kauai Dive Shoppe	Erica Norman	808-555-0269
1231	Unisoot	George Weathers	809-555-3915
1351	Sight Diver0	Phyllis Spooner	357-6-876708
1354	Cayman Divers World Unlimited	Joe Bailey	011-5-697044
1356	Tom Sawyer Diving Centre	Chris Thomas	504-798-3022
1380	Blue Jack Aqua Center	Ernest Barratt	401-609-7623
1384	VIP Divers Club	Russell Christopher	809-453-5976
1510	Ocean Paradise	Paul Gardner	808-555-8231
1513	Fantastique Aquatica	Susan Wong	057-1-773434
1551	Marmot Divers Club	Joyce Marsh	416-698-0399
1560	The Depth Charge	Sam Witherspoon	800-555-3798
1563	Blue Sports	Theresa Kunec	610-772-6704
1624	Makai SCUBA Club	Donna Slaus	317-649-9098
1645	Action Club	Michael Spurling	813-870-0239
1651	Jamaica SCUBA Centre	Barbara Harvey	011-3-697043
1680	Island Flinders	Desmond Ortega	713-423-5675
1984	Adventure Undersea	Gloria Gonzales	011-34-09054

### QUERY WIZARD

- 1 Click on the Data tab.
2. Select File | New in order to access the New Items dialog.
- 4 Double-click on the Query Wizard icon. The Query Wizard will come up with a list of Available Tables.
- 5 Choose the **Customer** Table by double-clicking on it. The table should now appear in the list of Selected Tables.
- 6 Click Finish.
- 7 Access the design workspace.

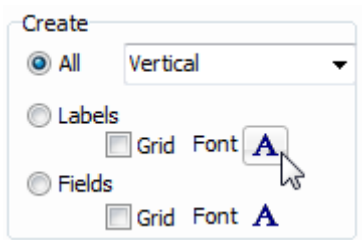
### DESIGN WORKSPACE

#### Task 1 - Launch the Data Tree and Set Properties

- 1 If it is not already visible, launch the Data Tree toolbar.
- 2 Dock it on the left side of the workspace.
- 3 Click the layout tab of the Data Tree.
- 4 Set the Style to Tabular.

5 Deselect the Grid boxes if they are checked.

6 Click the font icon for Labels.




7 Set the Label font to Arial, Bold, 10.

8 Set the Field font to Arial, Regular, 10.

9 Click on the data tab of the Data Tree.

10 Press Ctrl + S to save your work. Save the report as Summary Tutorial. Save at the end of each task.

## Task 2 - Lay Out Components in the Detail Band

1 Place your cursor over the 'Detail' divider. Your cursor will change to an up/down arrow , indicating that you can drag the control up and down.

2 Increase the height of the detail band by dragging the divider down until the guide reaches the one inch mark on the vertical ruler.

3 Place a shape component in the detail band.

4 Set the shape's fill color and line color to yellow.

5 Right-click over the shape and select ParentHeight and ParentWidth.

6 Hold down the Ctrl key while you select the '**Custno**', '**Company**', '**Contact**', and '**Phone**' fields in the Data Tree.

7 Drag the selection from the Data Tree into the lower left corner of the header band and release the mouse button. Labels should be created in the header band and DBTexts should be created in the detail band.



## Task 3 - Position the Components

1 Position the selection so that the labels line up with the bottom of the header band.

2 Deselect the components by clicking on the white space of the footer band.

3 Select all of the components in the detail band.



4 Shift-click the yellow shape to remove it from the selection.

5 Drag the selection up so that the components line up with the top of the detail band. The layout should look like this:



6 Deselect the components.

## Task 4 - Size the Components

1 Select the Company DBText component in the detail band.

2 Use the sizing handle to make the component narrower. Size the component so that the guide lines up with the three inch mark on the horizontal ruler.

3 Shift-click the Company label in the header band.

4 Launch the Size toolbar.

5 Locate the Shrink Width to Smallest icon  and click on it. The label's width should shrink to match that of the DBText.

## Task 5 - Align the Components and Adjust the Detail Band

1 Select the Contact and Phone labels.


2 Shift-click the corresponding DBText components in the detail band.

3 Drag the selection until it reaches the 3 1/8 inch mark on the horizontal ruler.

4 Left justify the DBText Custno component in the detail band.

5 Drag the detail band divider up until it meets the bottom of the components.

## Task 6 - Lay Out the Footer Band

1 Click the System Variable icon .

2 Click the lower left side of the footer band.

3 Select PrintDateTime from the drop-down list box in the upper left corner of the workspace. The date and time should appear in the component.

4 Place another system variable component in the lower right side of the footer band.

5. Select PageNoDesc (Page Number Description) from the drop-down list box. The page number should appear in the component.

6 Right justify the system variable by clicking the right-justify icon .

7 Align the tops of the system variables.

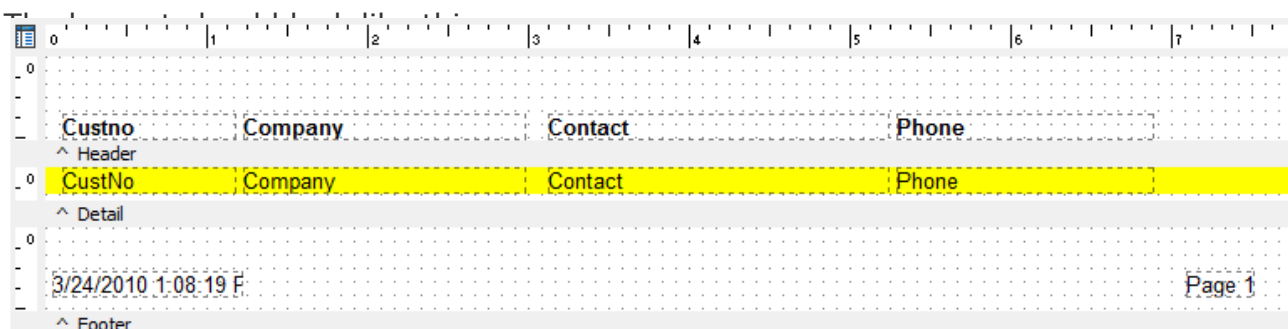
8 Press Ctrl + S to save your work.

## PREVIEW

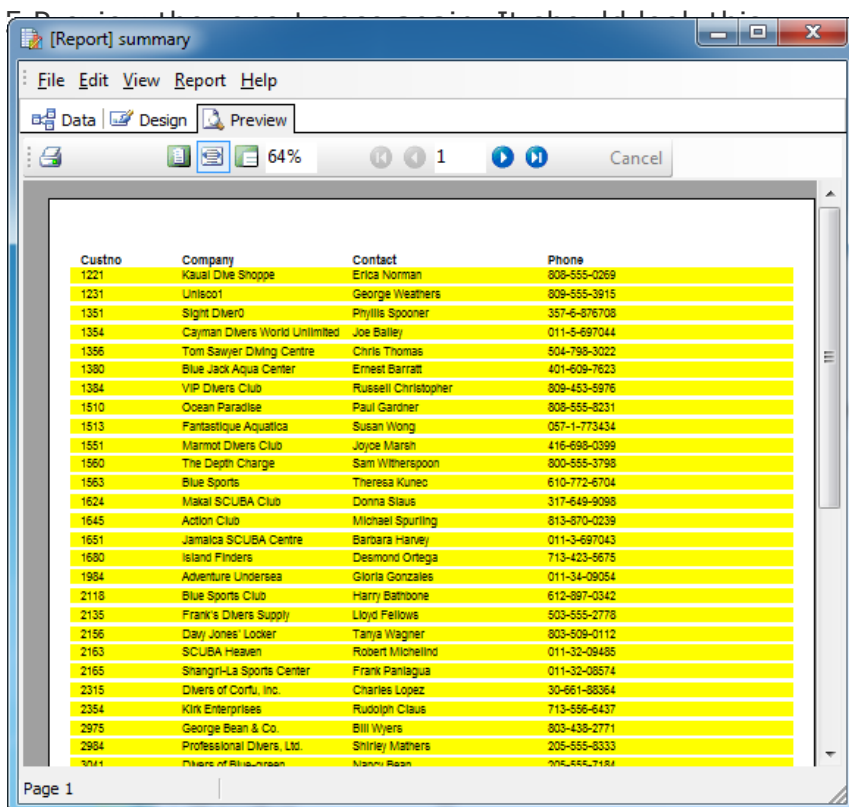
- 1 Click on the Preview tab and look at your report.
- 2 The data is solid yellow. Let's put white spaces between the data in order to differentiate the rows.

## FIXES

- 1 Click the Design tab to return to the design workspace.
- 2 Right-click over the yellow shape.
- 3 Select ParentHeight.
- 4 Drag the detail band divider down until you can see a little white space below the shape.



Custno	Company	Contact	Phone
1221	Kauai Dive Shoppe	Erica Norman	808-555-0269
1231	Uniscor1	George Weathers	809-555-3915
1351	Sight Diver0	Phyllis Spooner	357-6-676708
1354	Cayman Divers World Unlimited	Joe Bailey	011-5-697044
1356	Tom Sawyer Diving Centre	Chris Thomas	504-798-3022
1380	Blue Jack Aqua Center	Ernest Barratt	401-609-7623
1384	VIP Divers Club	Russell Christopher	809-453-5976
1510	Ocean Paradise	Paul Gardner	808-555-8231
1513	Fantastique Aquatica	Susan Wong	057-1-773434
1551	Marmot Divers Club	Joyce Marsh	416-698-0399
1560	The Depth Charge	Sam Witherspoon	800-555-3798
1563	Blue Sports	Theresa Kuneo	610-772-6704
1624	Makal SCUBA Club	Donna Slaus	317-649-9098
1645	Action Club	Michael Spurling	813-870-0239
1651	Jamaica SCUBA Centre	Barbara Hanley	011-3-697043
1680	Island Finders	Deemond Ortega	713-423-5675
1984	Adventure Undersea	Gloria Gonzales	011-34-09054
2118	Blue Sports Club	Harry Bathoone	612-897-0342
2135	Frank's Divers Supply	Lloyd Fellows	503-555-2778
2156	Clay Jones' Locker	Tanya Wagner	803-509-0112
2163	SCUBA Heaven	Robert Michelino	011-32-09485
2165	Shangri-La Sports Center	Frank Panlagua	011-32-08574
2315	Divers of Corfu, Inc.	Charles Lopez	30-661-88364
2354	Kirk Enterprises	Rudolph Claus	713-556-6437
2975	George Bean & Co.	Bill Wyers	803-438-2771
2984	Professional Divers, Ltd.	Shirley Mathers	205-555-8333
3041	Divers of Blue Ocean	Nancy Bean	205-555-7188



Custno	Company	Contact	Phone
1221	Kauai Dive Shoppe	Erica Norman	808-555-0269
1231	Uniscor1	George Weathers	809-555-3915
1351	Sight Diver0	Phyllis Spooner	357-6-676708
1354	Cayman Divers World Unlimited	Joe Bailey	011-5-697044
1356	Tom Sawyer Diving Centre	Chris Thomas	504-798-3022
1380	Blue Jack Aqua Center	Ernest Barratt	401-609-7623
1384	VIP Divers Club	Russell Christopher	809-453-5976
1510	Ocean Paradise	Paul Gardner	808-555-8231
1513	Fantastique Aquatica	Susan Wong	057-1-773434
1551	Marmot Divers Club	Joyce Marsh	416-698-0399
1560	The Depth Charge	Sam Witherspoon	800-555-3798
1563	Blue Sports	Theresa Kuneo	610-772-6704
1624	Makal SCUBA Club	Donna Slaus	317-649-9098
1645	Action Club	Michael Spurling	813-870-0239
1651	Jamaica SCUBA Centre	Barbara Hanley	011-3-697043
1680	Island Finders	Deemond Ortega	713-423-5675
1984	Adventure Undersea	Gloria Gonzales	011-34-09054
2118	Blue Sports Club	Harry Bathoone	612-897-0342
2135	Frank's Divers Supply	Lloyd Fellows	503-555-2778
2156	Clay Jones' Locker	Tanya Wagner	803-509-0112
2163	SCUBA Heaven	Robert Michelino	011-32-09485
2165	Shangri-La Sports Center	Frank Panlagua	011-32-08574
2315	Divers of Corfu, Inc.	Charles Lopez	30-661-88364
2354	Kirk Enterprises	Rudolph Claus	713-556-6437
2975	George Bean & Co.	Bill Wyers	803-438-2771
2984	Professional Divers, Ltd.	Shirley Mathers	205-555-8333
3041	Divers of Blue Ocean	Nancy Bean	205-555-7188

- 6 Close the Report Designer, saving the changes to the report.

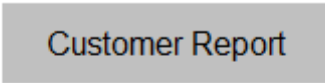
## Cover Page

### OVERVIEW

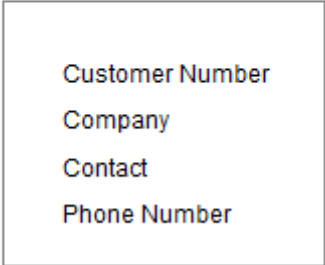
This tutorial shows you how to create a cover page for a report.

The final product will contain the following items:

- The title of the report
- A description of the contents of the report



Customer Report



Customer Number  
Company  
Contact  
Phone Number

### DESIGN WORKSPACE




#### Task 1 Create and Adjust the Title Band

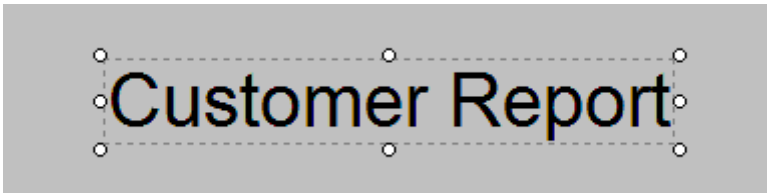
- 1 Open the Summary Tutorial report.
- 2 Select File | Save As from the main menu of the Report Designer.
- 3 Name the report Cover Page and click Save. The name at the top of the Report Designer should change to Cover Page.
- 4 Hide the Data Tree if it is visible.
- 5 Select Report | Title. A title band will appear at the top of the canvas.
- 6 Increase the height of the title band by dragging the divider down until the guide reaches the 9.5 inch mark on the vertical ruler.
- 7 Right click over the white space and select New Page.

#### Task 2 Create a Shape and a Label

- 1 Place a shape in the upper left corner of the title band.
- 2 Set the position of the shape:

Left 2.0  
Top 2.5  
Width 4.0  
Height 1

- 3 Set the fill and line color of the shape to light gray.
  - 4 Place a label over the shape.
  - 5 Set the caption to Customer Report.
  - 6 Set the font size to 28.
  - 7 Set the justification to left . 
  - 8 Select the shape and then shift-click the label.
  - 9 Launch the Align or Space toolbar.
  - 10 Click the Align Center icon  and the Align Middle icon  on the Align or Space toolbar.
- The label should be centered in the middle of the shape like this:



### Task 3 Create and Adjust a Shape

- 1 Place a new shape below the existing shape. Set the position of the shape:

Left 2.0  
Top 4.0  
Width 4.0  
Height 3.3

- 2 Set the line color of the shape to dark gray.

### Task 4 Create and Adjust Labels

- 1 Create four labels and place them over the shape.
- 2 Set the caption for each label:

Label2 Customer Number  
Label3 Company  
Label4 Contact  
Label5 Phone Number

- 3 Select the labels and set the font size to 22. The title page should look like this:




## Customer Report

Customer Number

Company

Contact

Phone Number

- 4 Select the Customer Report label and then shift-click the Customer Number label.
- 5 Click the Align Middle icon  of the Align or Space toolbar.
- 6 Set the top of the Customer Number label to 4.75.
- 7 Set the top of the Phone Number label to 6.5.
- 8 Select the Customer Number label and then shift-click the three labels below it.
- 9 Click the Left Align icon  and the Space Vertically  icon.

### PREVIEW

- 1 Click the Preview tab. The title page should look like this:

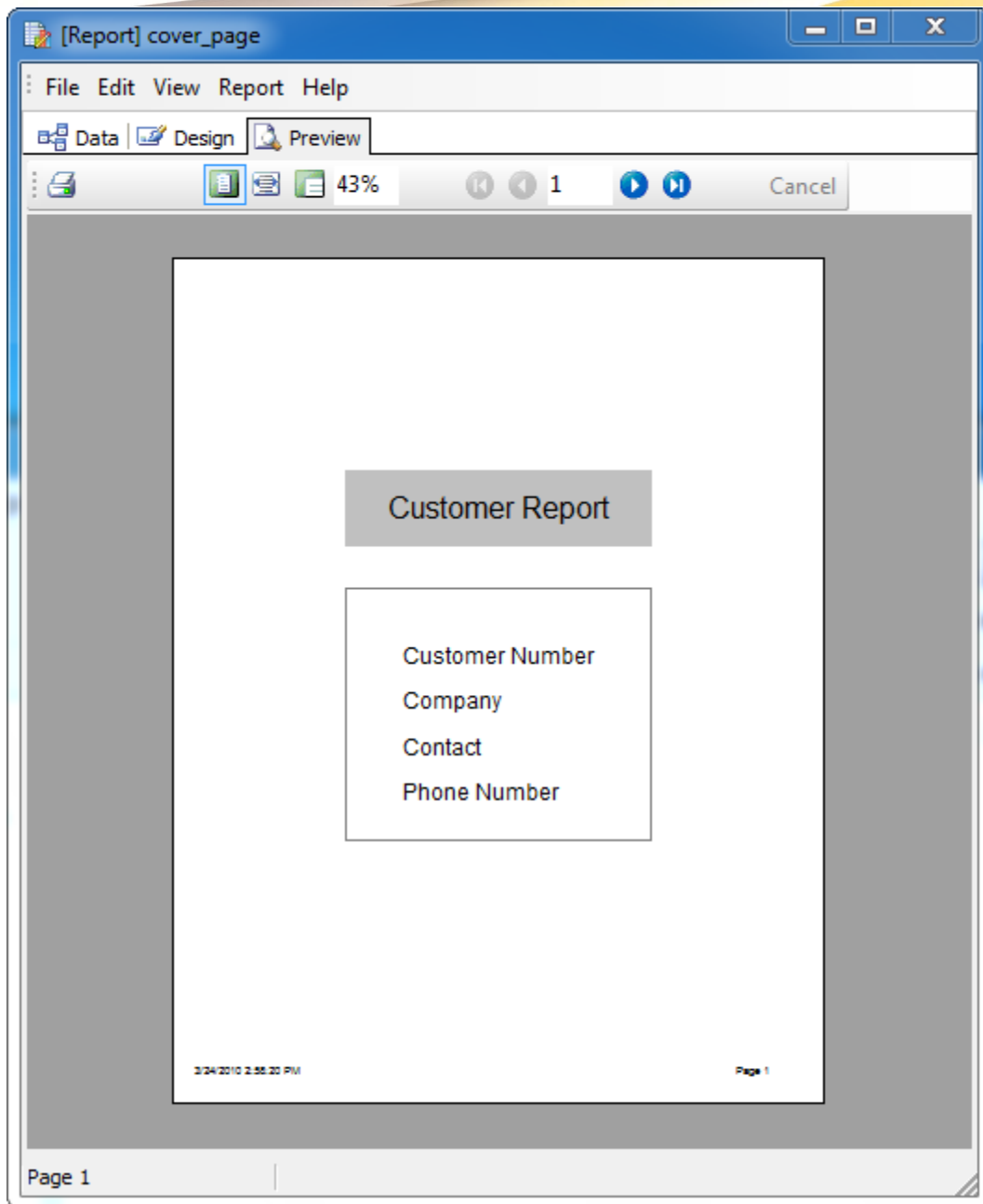
## Customer Report

Customer Number

Company

Contact

Phone Number



- 2 Click the Next Page button. The rest of the Summary Report should appear.
- 3 Close the Report Designer and save the changes.

## Simple Table Listing

### OVERVIEW

This tutorial illustrates how to create a table style report.

The final product will include the following items:

- A list of company names
- A contact for each company
- A list of orders for each company
- The amount paid for each order

#### *Sales Detail by Company*

Company	Contact	Orderno	Amountpaid
Action Club	Michael Spurling	1014	134.88
		1029	20108
		1038	10152
Action Over Supply	Marianne Miles	1039	538.8
Adventure Undersea	Gloria Gonzales	1017	89
		1027	3117
Blue Glass Happiness	Christine Taylor	1042	971.7
Blue Jack Aqua Center	Ernest Small	1008	548
		1053	
Blue Sports	Theresa Kunc	1012	5201
Blue Sports Club	Harry Balbone	1018	48
		1031	12885
Davy Jones' Locker	Tanya Wagner	1004	52
		1020	9958

### QUERY WIZARD

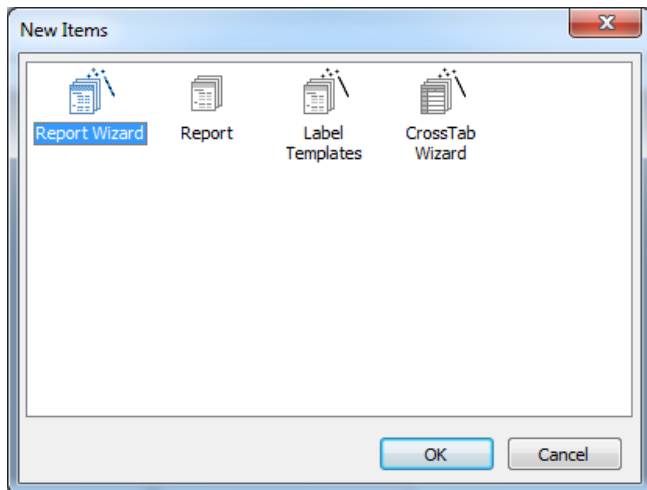
#### Task 1 Access the Query Wizard and Select Fields

- 1 Create a new report.
- 2 Access the data workspace.
- 3 Select File | New.
- 4 Double-click on the Query Wizard icon.
- 5 Double-click on the Customer table in order to select it.
- 6 Double-click on the Orders table.
- 7 Click Next until you reach the screen with the Set Order option.
- 8 Click Set Order and select Customer.Company, Customer.Contact, and Orders.Orderno.
- 9 Click Finish. The completed database will appear.
- 10 Access the design workspace.

## DESIGN WORKSPACE

### Task 1 Create a Layout Via the Report Wizard

- 1 Select File | New in order to access the Report Wizard.
- 2 Double-click on the Report Wizard Icon.



- 3 Select the following fields by double clicking:

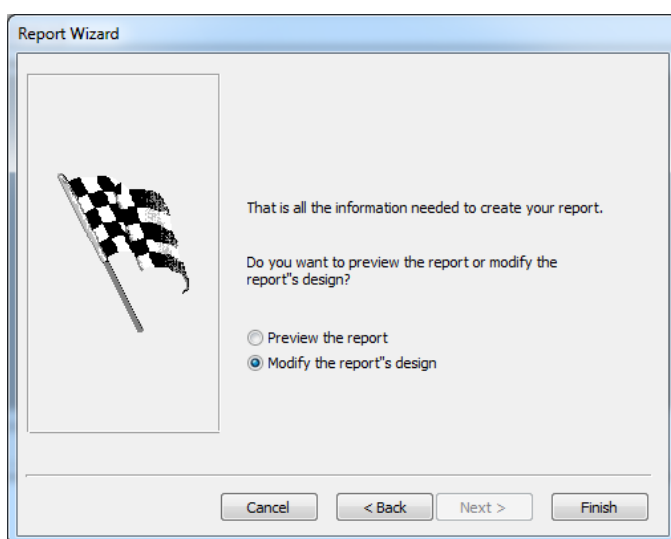
Company

Contact

Orderno

Amountpaid

- 4 Click Next until with the checkered racing flag. Select 'Modify the report's design'.





5 Click Finish. The report layout will appear:



The screenshot shows the Navicat report design workspace. At the top is a ruler. Below it is a report titled "Sales Detail by Company" in a blue, italicized font. The report is divided into several bands: Title, Header, Detail, and Footer. The Header band contains four columns: "Company", "Contact", "OrderNo", and "AmountPaid". The Detail band contains a single row with the same four columns. The Footer band contains a date "3/24/2010 3:0" and a page number "Page 1".

6 Right-click over the New Report label in the header band and select Autosize from the speed menu.

7 Select the New Report label and set the caption to 'Sales Detail by Company'.

8 Press Ctrl + S and save the report as Simple Table.

9 Click the Preview tab. Notice the repetition of the company and contact names: this repetition inhibits readability. Let's make the report more readable by showing these values only once.

## Task 2 Select SuppressRepeatedValues


1 Return to the design workspace.

2 Right-click over the Company DBText component in the detail band and select SuppressRepeatedValues from the speed menu.

3 Right-click over the Contact DBText component in the detail band and select SuppressRepeatedValues.

4 Click the Preview tab. The company and contact names no longer repeat.

5 Scroll down to the bottom of the preview page. Notice that the last company name is Blue Sports Club.

6 Click the Next Page icon . Notice that 'Blue Sports Club' does not appear at the top of this page. Because the company and contact names do not repeat, the top of the next page has no company name; it is therefore difficult to tell what company placed these orders. Let's modify the report so that the company and contact names reprint at the top of new pages.

## Task 3 Select ReprintOnSubsequent

1 Return to the design workspace.

2 Right-click over the Company DBText component and select ReprintOnSubsequent.


3 Right-click over the Contact DBText component and select ReprintOnSubsequent.

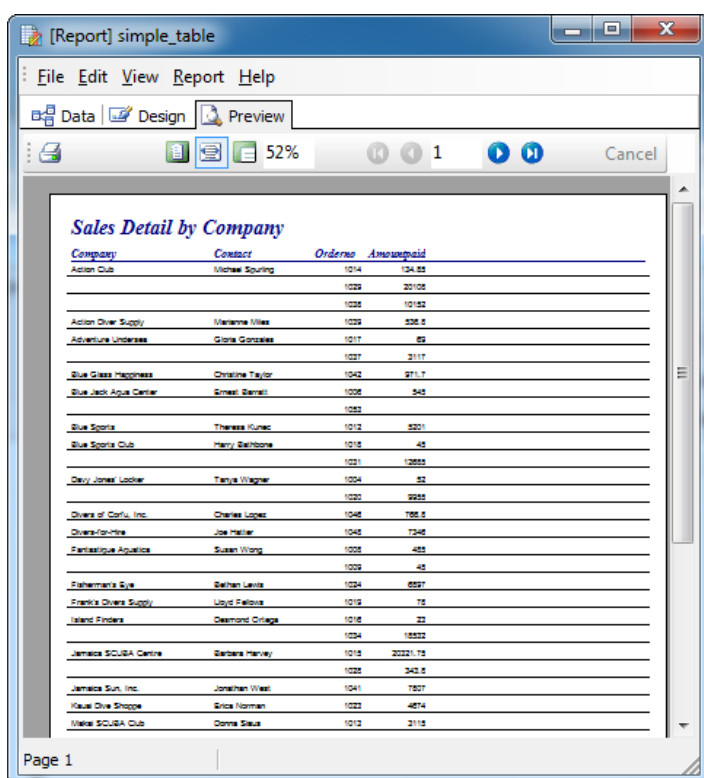
4 Click the Preview tab.

5 Go to the second page. Notice that 'Blue Sports Club' is now at the top of the page, providing clarification.

## Task 4 Create a Line Guide

You may have noticed that the ReprintOnSubsequent option creates a lot of white space in this report, which makes it difficult to correlate company names with their orders as you read from left to right. We can add a line guide in order to improve this situation.

- 1 Return to the design workspace.
- 2 Click on the line icon  on the Standard component palette.
- 3 Click in the detail band to create the line component.
- 4 Right-click over the line and select ParentWidth from the speed menu. The line will stretch to match the width of the band.
- 5 Set the Top of the line to 0.15.
- 6 Press Ctrl + S to save your work.
- 7 Access the preview screen. The completed report should look like this:



- 8 Close the Report Designer.

## Invoice

### OVERVIEW

This tutorial shows you how to create an invoice. The goal of this exercise is to illustrate groups and their function.

The final product will present a set of invoices. Each invoice will contain the following items:

- A company name
- A contact for each company
- The orders for each company
- The amount paid for each order
- The total amount paid for each company

### *Invoice*

<i>Company</i>	<i>Contact</i>	<i>Order No.</i>	<i>Amount Paid</i>
Action Club	Michael Spurling		
		1014	134.85
		1029	20108
		1038	10152
			\$30,394.85

## ESTABLISH THE BASIC DESIGN

### Task 1 Use the Dataview from the Simple Table Report

- 1 Open the Simple Table report.
- 2 Access the data workspace.
- 3 Select File | Export. A save dialog box will appear.
- 4 Create a folder named Data and open it.
- 5 Name the data 'Customer Orders' and click save.
- 6 Close the Report Designer.
- 7 Click on the New Report icon.
- 8 Access the data workspace.
- 9 Select File | Import.
- 10 Double-click on Customer Orders. You should now see the Customer Orders dataview in the data workspace.

Note: All of the dataviews in the data workspace are stored in what is called a data module. You can use the export feature to save a data module outside of a report.

When you find yourself using the same dataviews over and over again, it is often useful to save them in a data module outside of the report so that you can use them (via the import feature) when creating new reports.

## Task 2 Generate the Layout Via the Report Wizard

1 Access the design workspace.

2 Select File | New and double-click on the Report Wizard icon.

3 Select the following fields:

Company

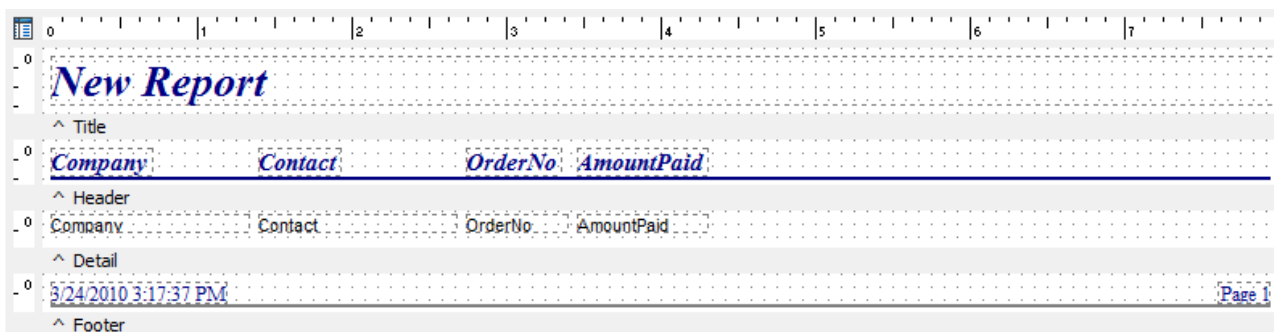
Contact

Orderno

Amountpaid

4 Click Next until you reach the screen with the checkered racing flag. Select 'Modify the report's design'.

5 Click Finish. The report layout will appear:



6 Select the New Report label and set the caption to Invoice.

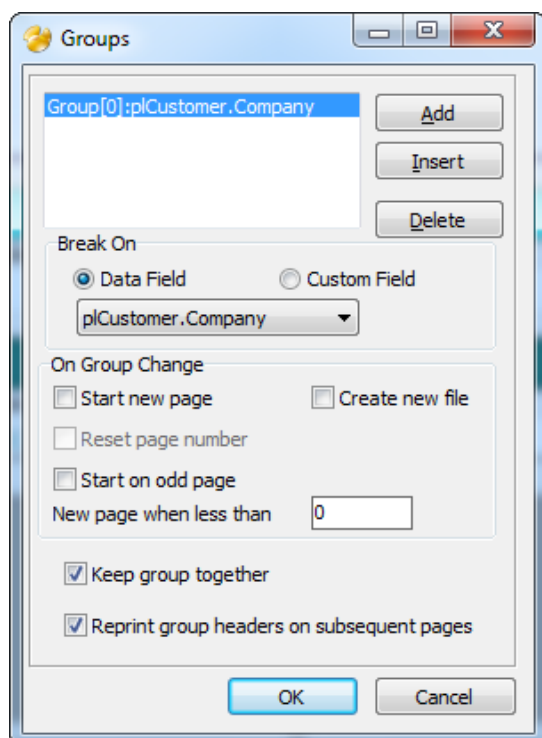
7 Press Ctrl + S and select 'All Folders' from the drop-down list. Set the name of the report to Invoice and click on the save button.

8 Preview the report. Notice how the company and contact names repeat. In the previous tutorial, we learned how to handle repeated values via the SuppressRepeatedValues property. However, we can handle these values another way.

## Task 3 Create a Group

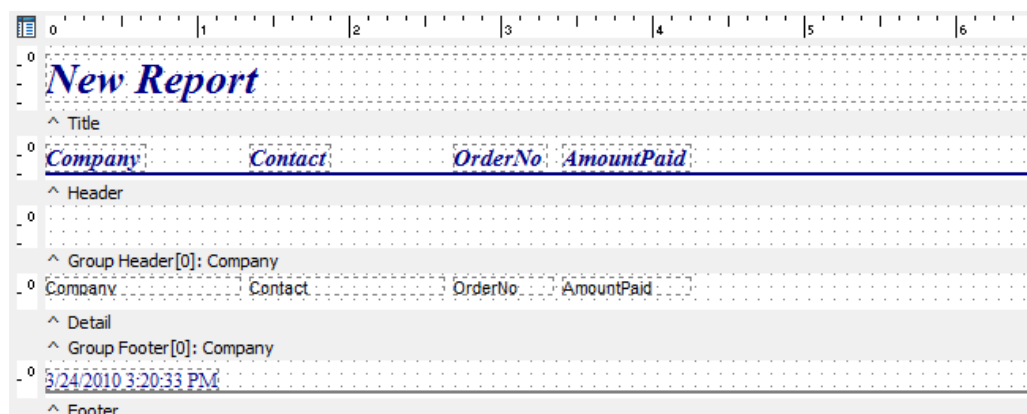
1 Return to the design workspace and select Report | Groups from the main menu. A dialog box will appear.

2 Choose Customer.Company from the drop-down list box:



3 Click Add, then click OK. A group header band and a group footer band will appear on the canvas.

4 Place your cursor over the gray area labeled Group Header and drag down. White space for the group header band will be created.



5 Right-click over the white space and set the height of the band to 0.3.

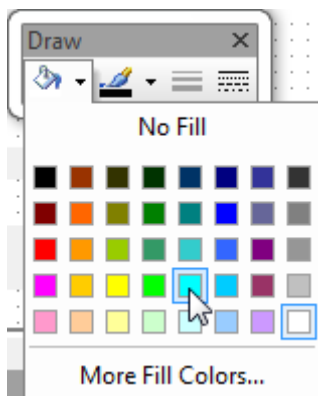
6 Right-click over the group footer band and set the height to 0.3.

## Task 4 Lay Out the Group Bands

1 Place a shape in the group header band.

2 Set the shape to ParentHeight and ParentWidth.

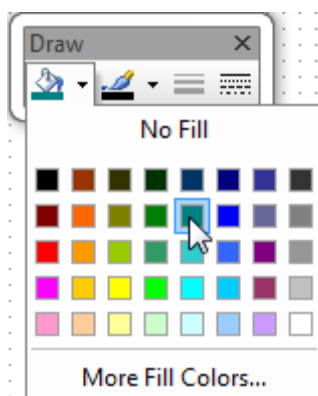
3 Set the shape's color to aqua.



4 Place a shape in the group footer band.

5 Set the shape to ParentHeight and ParentWidth.

6 Set the shape's color to teal (the color above aqua on the fill color palette):




7 Preview the report. The aqua shapes show where the group header band is printing; the teal shapes show where the group footer band is printing. Everything you see that begins with an aqua shape and ends with a teal shape is a group. A group, then, includes the following bands: the group header, detail, and group footer. Here is an example of how a single group looks on the preview page:

Action Club	Michael Spurling	1014	134.85
Action Club	Michael Spurling	1029	20108
Action Club	Michael Spurling	1038	10152

## Task 5 Move the Repeating Data into the Group Header



As you preview, notice that the company and con-tact fields still repeat. This is because they are in the detail band. We can fix this by moving these fields into the group header band.

1. Return to the design workspace.
- 2 Select the DBText components for the Company and Contact fields in the detail band.
- 3 Drag the selection into the group header band.
- 4 Right-click over the shape and select Send to Back. The DBText components should now appear in front of the shape.
- 5 Select the shape and then shift-click the DBText components. Click the Align Center icon  on the Align or Space toolbar. The components will be centered vertically in the shape.
- 6 Preview the report. The company and contact names should now appear in the group header, printing only once for each company.

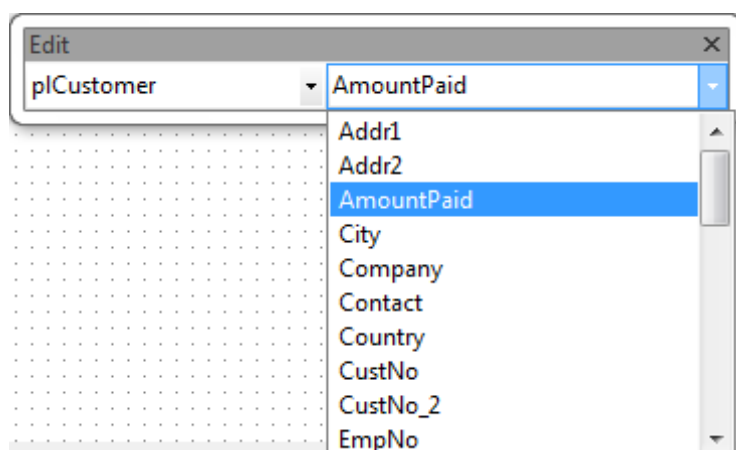
# Invoice

Company	Contact	Order No.	Amount Paid
Action Club	Michael Spurling		
		1014	134.85
		1029	20108
		1038	10152
Action Diver Supply	Marianne Miles		

## Task 6 Total the Amount Paid for Each Company

- 1 Return to the design workspace.
- 2 Select the Orderno DBText component in the detail band. This initializes the font style and size for the DBCalc component we are about to create.
- 3 Click the DBCalc icon , then click in the group footer band. A DBCalc component will be created.
- 4 Change the font color to white.
- 5 Click the right justify icon .

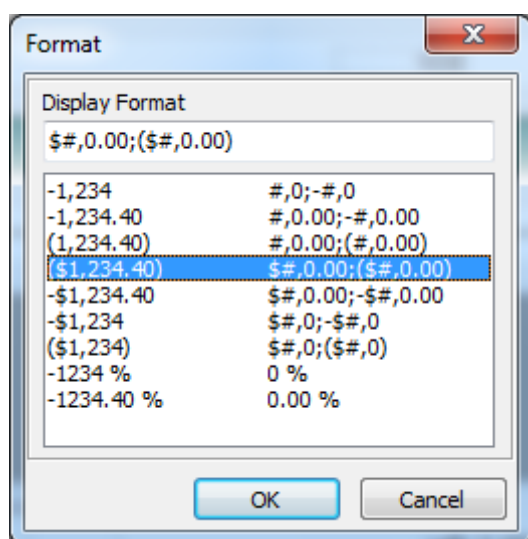
6 Select the 'Amountpaid' field from the edit tool-bar:



This tells the DBCalc component to sum the amount paid for each company.


7 Right-click over the DBCalc component and select AutoSize.

8 Right-click and select DisplayFormat. Click on the first format that begins with a dollar sign:



9 Click OK.

10 Adjust the DBCalc component so that it aligns with the bottom of the group footer band.

11 Click the OrderNo DBText in the detail band, then shift-click the DBCalc component. Click the Align Right icon  on the Align or Space tool-bar.



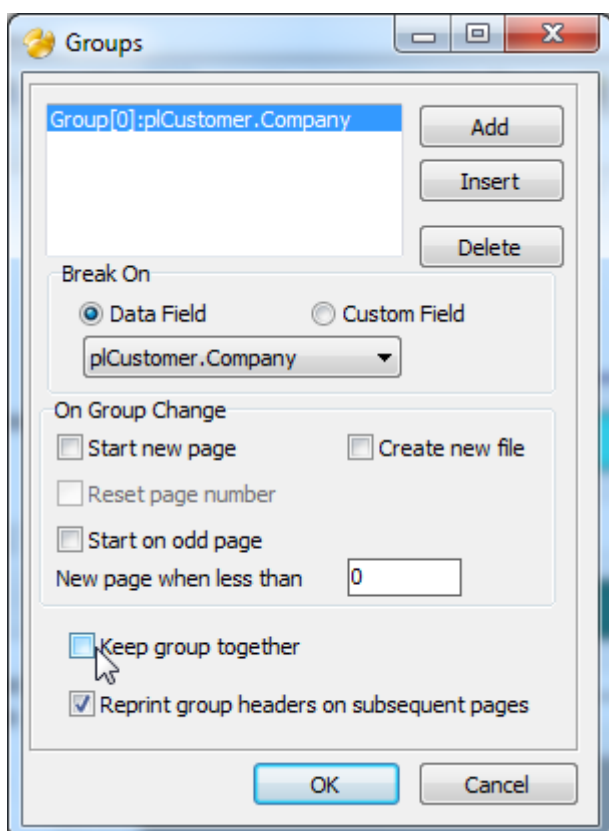
12 Preview the report. Notice that we now have a total in the group footer band.

*Invoice*

<i>Company</i>	<i>Contact</i>	<i>Order No.</i>	<i>Amount Paid</i>
Action Club	Michael Spurling		
		1014	134.85
		1029	20108
		1038	10152
			\$30,394.85
Action Diver Supply	Marianne Miles		
		1039	536.8
			\$536.80

## Task 7 Control the Pagination of the Groups

- 1 Return to the design workspace.
- 2 Select Report | Groups and uncheck 'Keep group together'.

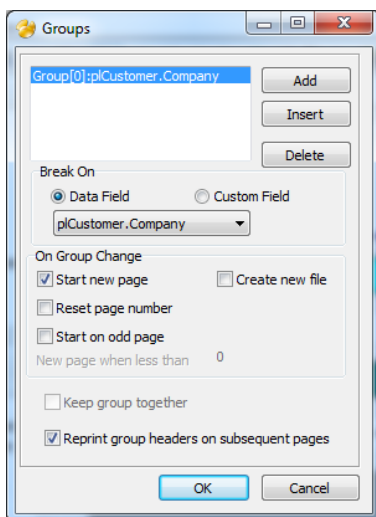


- 3 Scroll to the bottom of the first preview page. Notice how there is no total for the 'Blue Jack Aqua Center'. You must go to the next page in order to see the remaining detail lines and total for this group. Let's modify the report so that groups will not break across pages.
- 4 Return to the design workspace.
- 5 Select Report | Groups from the main menu.
- 6 Click the 'Keep group together' option:
- 7 Click OK.
- 8 Preview the report. The group for 'Blue Jack Aqua Center' now starts on page two. The 'Keep group together' option allows the header, detail, and footer of a group to print on the same page, when possible.

## Task 8 Begin Each Group on a New Page

Even though 'Keep group together' improves the report's pagination, we need invoices that can be sent to each individual company. This means that information from multiple companies cannot appear on the same page. Let's modify the report to handle this requirement.

- 1 Return to the design workspace.
- 2 Select Report | Groups. Click 'Start new page':



Notice that the 'Keep group together' option is dis-abled. This option is no longer necessary because each group will begin on its own page; thus, each group has the maximum space to print.

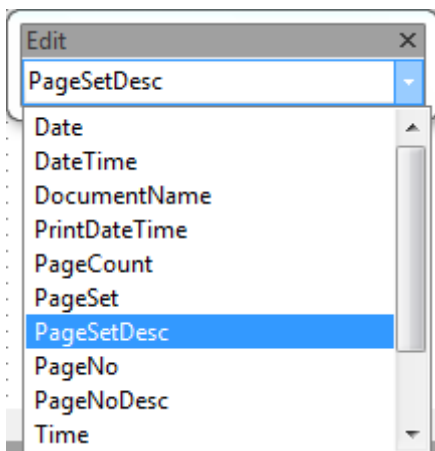
- 3 Click OK.
- 4 Return to the preview screen and confirm that each group starts on its own page.
- 5 Select Ctrl + S to save your work.

## Task 9 Number the Pages by Company

As you preview, notice how the pages are num-bered. Each company will receive only one or two

pages of the report. Therefore, it would not be acceptable to send out an invoice that says 'Page 37' at the bottom. We can fix this problem by using the Reset Page Number feature of groups.

- 1 Return to the design workspace.
- 2 Select the Page 1 System Variable in the footer band.
- 3 Select PageSetDesc (Page Set Description) from the edit toolbar:



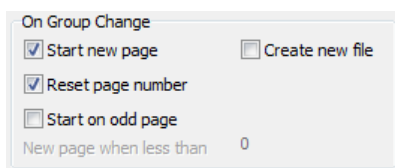
- 4 Select Report | Groups.
- 5 Check the Reset page number option:
- 6 Click OK.
- 7 Preview the report. The page numbers should reset for each company. Each invoice is one page, so the page number displays 1 of 1.

## FIT AND FINISH

At this point, the report is functioning well. We just need to give it a professional look and feel.

### Task 1 Adjust the Line Component in the Header Band

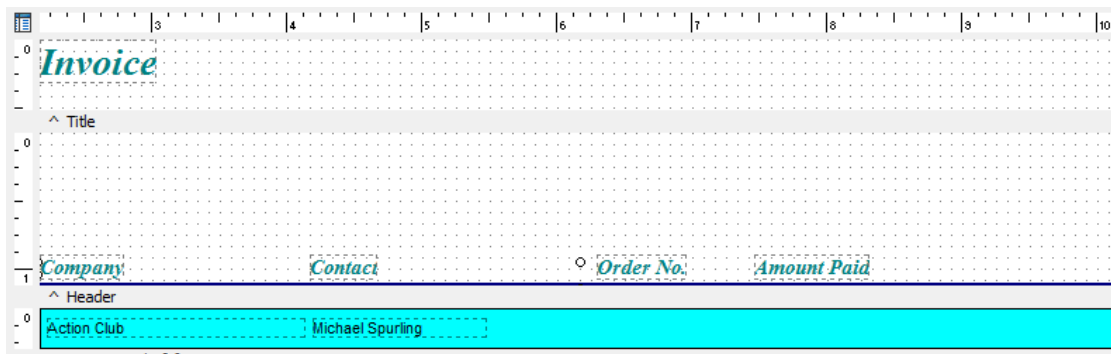
- 1 Return to the design workspace.
- 2 Select the line in the header band and select bot-tom from the list in the edit toolbar.



- 3 Extend the header band to 1 1/8 inch.

4 Right-click over the line and set it to Parent-Width.

5 Drag the line to the bottom of the header band.



## Task 2 Adjust the Label Components in the Header Band

1 Right-click over the Contact label and set the top to 0.9.

2 Align the tops of all the labels in the detail band with the Contact label.

3 Set the following left positions for the labels:

Company 0.0

Contact 2.0

Orderno 4.2

Amountpaid 5.3

4 Change the captions of the labels as follows:

Orderno Order No.

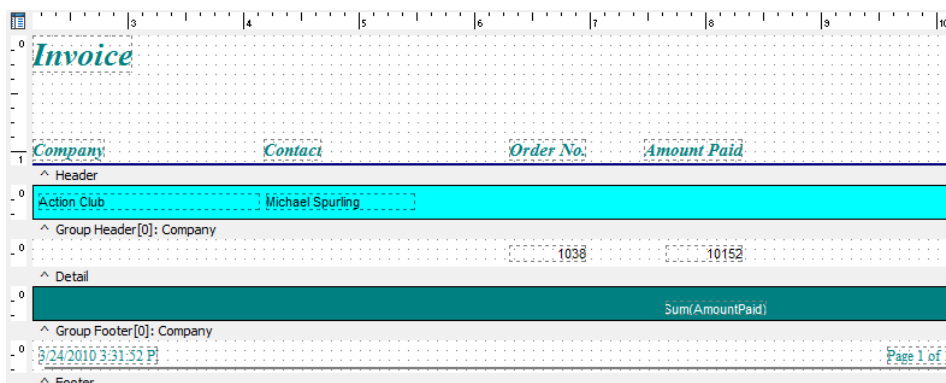
Amountpaid Amount Paid

5 Set each of the labels to AutoSize.

## Task 3 Align the Components

1 Select the Amount Paid label.

2 Shift-click the corresponding DBText and DBCalc components in the detail and group footer bands.



3 Use the Align or Space toolbar to right-align the selection.

4 Right-align the Order No. DBText component in the detail band with the Order No. label in the header band.

5 Set the top of both DBText components in the detail band to 0.05.

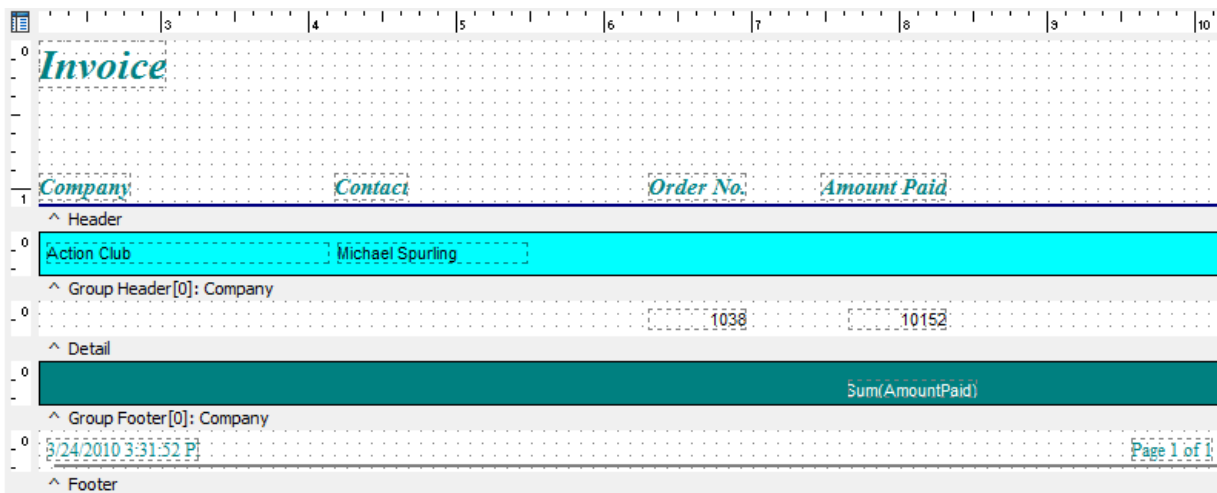
## Task 4 Modify the Label in the Title Band

1 Drag the label in the title band into the header band.

2 Set the left and top of the label to 0.

3 Select Report | Title from the main menu. The title band will be removed from the canvas.

4 Right-click over the label and select AutoSize. The layout should look like this:



5 Press Ctrl + S to save your work.

6 Preview the report.

## Task 5 Change the Color of the Navy Components

Everything looks great except the components in the title, header, and footer bands, which are navy. Let's change them to match the rest of the report.

1 Return to the design workspace.

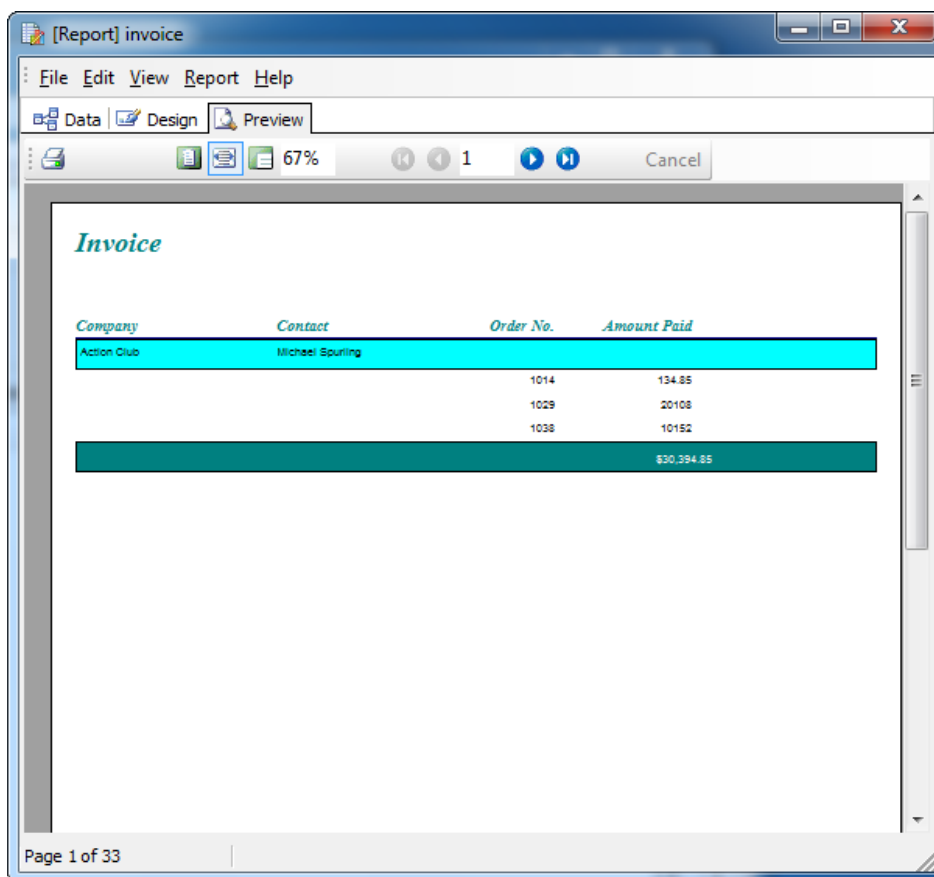
2 Use the shift-click method to select all navy components.

3 Change the font color to teal using the Fill Color palette.

4 Change the color of the line in header band to teal using the Line Color palette.

5 Preview the report. The first page should look like this:

6 Close the Report Designer and save the changes.



## Order Summary

### OVERVIEW

This tutorial demonstrates how to create an order summary. The goal of this exercise is to demonstrate how to link dataviews to create a master-detail relationship; and to illustrate the use of a subreport to create a master-detail report. The final product will present an order summary for multiple companies and their multiple orders. Each summary will include the following items:

- A company name
- A customer number
- An order number for each individual order
- The total amount owed per order
- The total amount owed per company

### ***Action Club*** **1645**

<i>Order</i>	<i>Amount Due</i>
<i>1038</i>	<i>\$10,152.00</i>
<i>1029</i>	<i>\$20,108.00</i>
<i>1014</i>	<i>\$134.85</i>
<i>Total Amount Due</i>	<i>\$30,394.85</i>

## DEFINE A MASTER-DETAIL RELATIONSHIP

### Task 1 Create the Dataviews

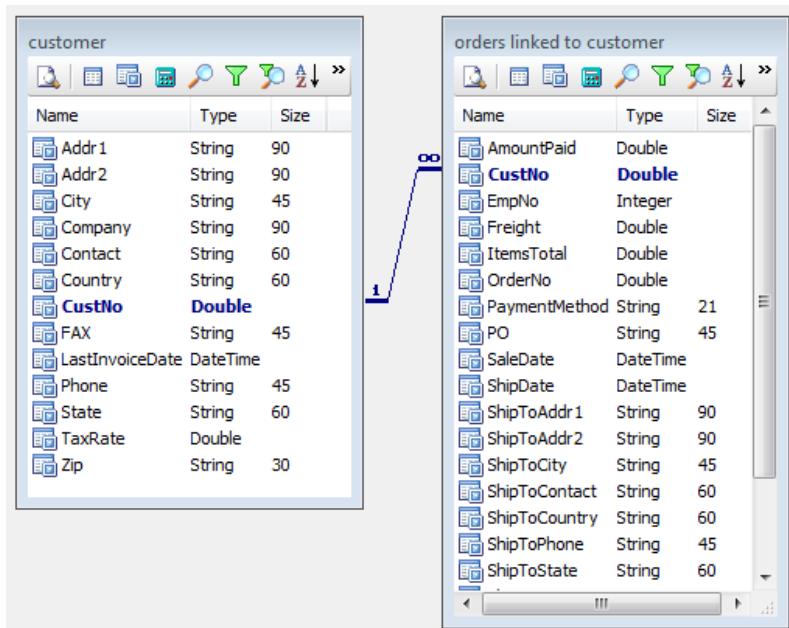
- 1 Create a new report.
- 2 Access the Query Wizard.
- 3 Select the Customer table.
- 4 Click Finish.
- 5 Select the Sort icon on the Customer dataview.
- 6 Double-click the Company field. The Company field is moved from Available Fields to Sort

Fields.

7 Click OK.

8 Repeat steps 1-4 to create a dataview for the Orders table.

Note: Notice the link between the dataviews. This is an AutoLink; it expresses a master-detail relationship. Master-detail relationships are one-to-many associations between the rows of the two datasets. See the below for more on this.



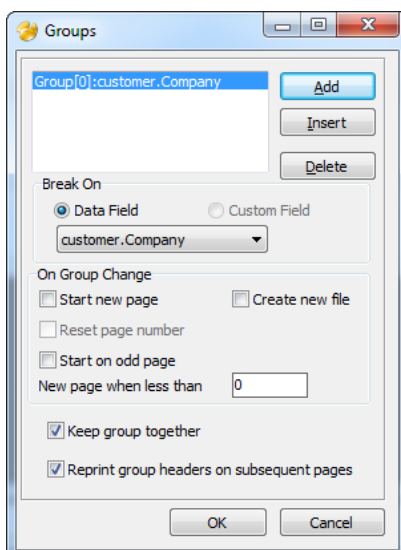
## Task 2 Create a Group

1 Go to the design workspace.

2 Select Report | Groups from the main menu. A dialog box will appear.

3 Choose Customer.Company from the drop-down menu.

4 Click the Add button. A group based on the company field will appear in the list box.





5 Check Start New Page.

6 Click OK.

You have created a group which includes the following bands: group header, detail, and group footer. This group will track the value of the company field and start a new page each time the value of that field changes.

7 Select Report from the main menu and deselect the header band. The header band will be removed from the layout.

8 Press Ctrl + S and save the report under the name Order Summary.

### **Task 3 Lay Out the Group Header Band**

1 Place a DBText component in the group header band and select Company from the drop-down list box.

2 Set the Company DBText component proper-ties:

AutoSize True

Left 0.0

Top 0. 0

Font Name Arial

Font Size 16 pt.

Font Style Bold and Italic

3 Place another DBText component in the group header band and select CustNo from the drop-down list box.

4 Set the CustNo DBText component properties:

Left 0.0

Top 0. 3

Font Name Arial

Font Size 16 pt.


Font Style Bold and Italic

5 Drag the group header band up until it meets the bottom of the CustNo DBtext component.

6 Click the Preview tab.

As you step through the pages, notice how each company starts on a new page. This is due to the Start New Page option of the group.


## Task 4 Lay Out the Footer Band

- 1 Click the design tab to return to the design workspace.
- 2 Place a system variable  in the upper-left corner of the footer band.
- 3 Select PrintDateTime from the drop-down list box.
- 4 Place another system variable in the upper-right corner of the footer band.
- 5 Select PageNoDesc from the drop-down list box.
- 6 Right justify the system variable.
- 7 Select both system variables.
- 8 Set the font:

Name Arial  
Size 10 pt.  
Style Bold

- 9 Click the Align Top icon to give both variables the same top.




- 10 Place a line component  in the footer band.
- 11 Set the Line component properties:

ParentWidth True  
Top 0.2

- 12 Drag the footer band up until it meets the bottom of the line component.

## MODIFY THE BASIC REPORT LAYOUT

### Task 1 Create a Subreport

- 1 Place a subreport component  in the detail band.
- 2 Select the Orders dataview from the drop-down list box.
- 3 Right-click the subreport and select Position.
- 4 Set the top to 0.5.
- 5 Drag the detail band up to meet the bottom of the subreport.
- 6 Select the tab titled 'SubReport1 : Orders' at the bottom of the data workspace.

## Task 2 Create a Group in the SubReport

- 1 Select Report | Groups.
- 2 Choose Orders : CustNo from the drop-down menu.
- 3 Click Add.
- 4 Click OK.
- 5 Select Report from the main menu and deselect the title and summary bands.

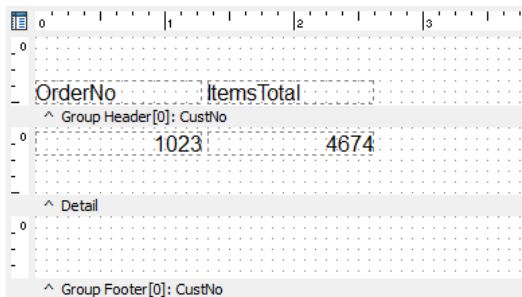
## Task 3 Use the Data Tree to Create Components in the SubReport

- 1 Launch the Data Tree. If it does not appear docked on the left side of the Report Designer, dock it there.
- 2 Select the Layout tab in the Data Tree.
- 3 Set the style to Tabular.
- 4 Select the Data tab and Ctrl-click the following fields:

OrderNo

ItemsTotal

- 5 Drag the selection to the bottom left corner of the group header band. Components should be created in the group header and detail bands.



- 6 Change the caption of the ItemsTotal label to Amount Due'.
- 7 Set the properties of both labels:

Top 0. 0

Font Color Navy Blue

- 8 Set the OrderNo label to Autosize.

- 9 Set the DBText components properties:

Top 0. 25

Font Name Times New Roman

Font Size 11 pt.

Font Style Bold and Italic

Font Color Green layout check

10 Set the Amount Due label and the ItemsTotal DBText components properties:

Left 2.0

Alignment Right Justify

11 Drag the group header band up to meet the bottom of the labels.

12 Drag the detail band up to meet the bottom of the DBText components.

## **Task 4 Calculate the Total Amount Due**

1 Place a label component in the group footer band and set the properties:

Caption Total Amount Due

Left 0.0

Top 0. 25

Font Name Arial

Font Size 11 pt.

Font Style Bold and Italic

Font Color Navy Blue

2 Place a DBCalc component in the group footer band. Select ItemsTotal from the drop-down list box.

3 Right-click the DBCalc component and select Calculations.

4 Select the Calc type Sum and click OK.

5 Right align the Amount Due Label in the group header band and the DBCalc component in the group footer band.

6 Set the DBCalc properties:

Autosize True

Alignment Right Justify

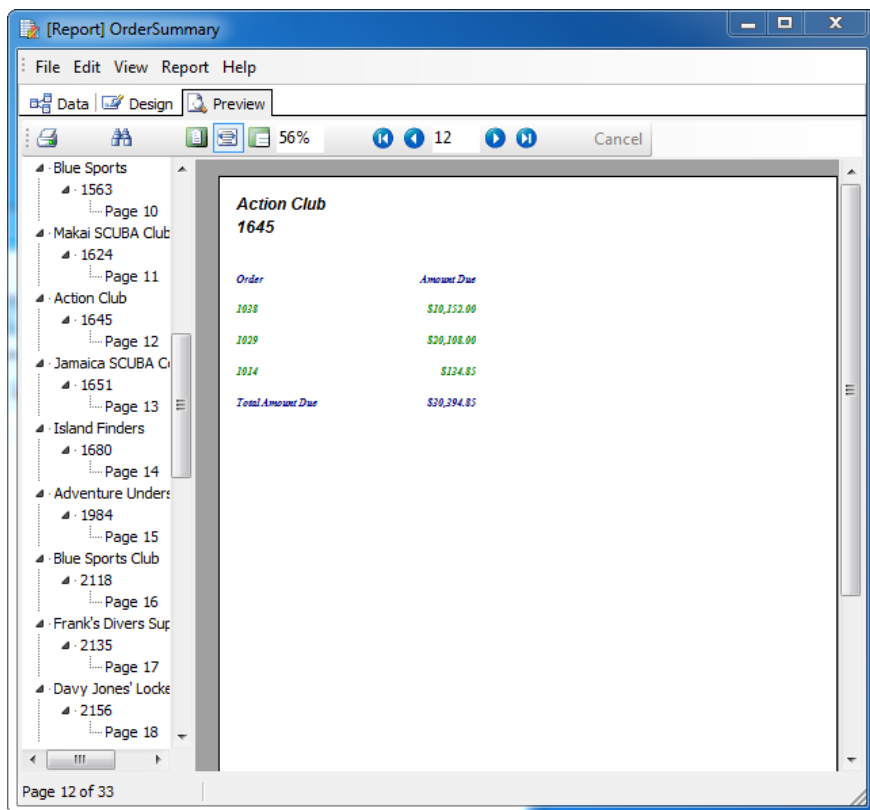
7 Right-click the DBCalc component and select Display Format. Choose the first format with a dollar sign.

8 Align the tops of the Total Amount Due label and the DBCalc component by shift-clicking both components and using the Align Top icon.

9 Set the height of the group footer band to 0.75.

## Task 5 Preview and Finish

- 1 Click the Preview tab.
- 2 As you preview, notice how the total amount due is the sum of the orders for each company.



- 3 Close the Report Designer, saving the changes. You have created an order summary with a master-detail relationship between dataviews with a subreport showing the detail data

## Detailed Order Summary

### OVERVIEW

In this tutorial we will learn how to create a master-detail relationship using the AutoLink feature; how to create a join between two tables within a data-view; and how to create a report containing multi-ple SubReports.

The final product will include the following items:

- a company name
- an order number for each order
- an item number for each item
- the quantity of each item
- the price of each item
- the total amount for each item

***Fantastique Aquatica***  
**1513**

<b>Order 1009</b>				
ItemNo	PartNo	Qty	Discount	
1	1320	7	0.00 %	

<b>Order 1008</b>				
ItemNo	PartNo	Qty	Discount	
2	11035	10	0.00 %	
1	1313	1	0.00 %	

## BUILDING ONTO THE REPORT LAYOUT

### Task 1 Adding More Detailed Data

- 1 Open the Order Summary Report.
- 2 Select File | Save As from the main menu of the Report Designer.
- 3 Name the report Detailed Order Summary and click Save. The name at the top of the Report Designer should be updated.
- 4 Access the Data tab.
- 5 Access the Query Wizard.
- 6 Select the Items table .
- 7 Click Finish.
- 8 Drag from OrderNo in the Items dataview to OrderNo in the Orders dataview. This is an example of creating a link to form a master-detail

relationship.

9 Select the Tables  icon on the Items data-view.

10 Double-click on the Parts table. Notice that Parts moves from the Available Tables box to the Selected Tables box.

11 Select the Fields tab.

12 Check the All Fields box.

13 Click OK.

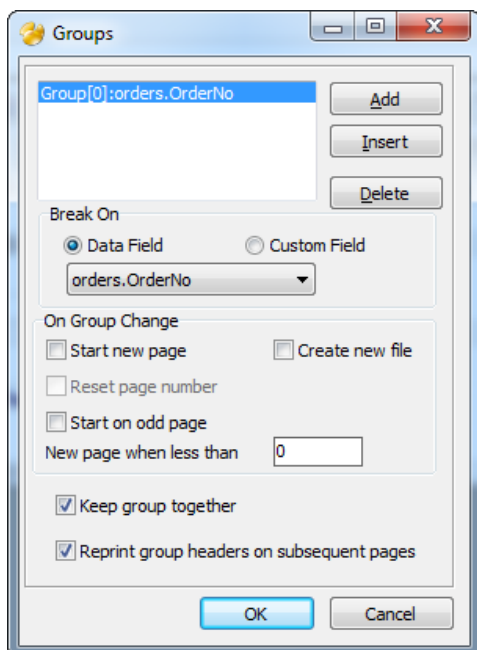
## Task 2 Change the Group

1 Return to the Design tab.

2 Select the tab titled SubReport1: Orders the bottom of the data workspace.

3 Select Report | Groups.

4 Select Groups and set field to Orders.OrderNo.



5 Click OK.

## Task 3 Layout of the Group Header and Footer Bands

1 Delete the Amount Due label in the group header band.

2 Create a DBText component in the group header band and select OrderNo from the drop-down list.

3 Set the DBText component properties:

Autosize True

Left 0.6

Top 0. 2

Font Name Arial

Font Size 11

Font Style Bold and Italic

4 and align the top with the OrderNo DBText component.

5 Create a line component in the group header band

6 Set the line component properties:

Left 0

Top 0. 2

Height 0.2083

Font Color Navy Blue

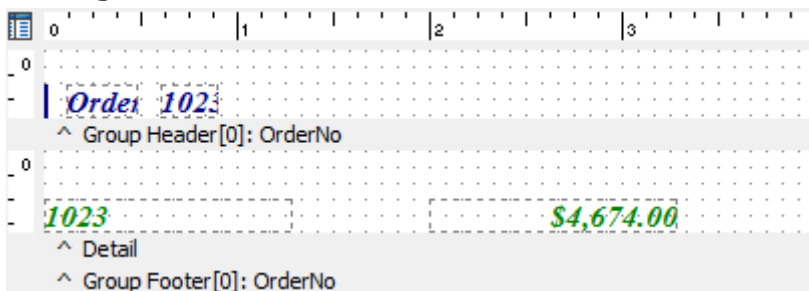
Line Thickness 2 pt.

Line Position Left

7 Drag the group header band up to meet the bottom of the components.

8 Delete everything in the group footer band.

9 Drag the group footer band up to meet the detail band so that none of the white space is showing.



## Task 5 Create a SubReport

1 Delete the OrderNo and ItemsTotal DBText components in the detail band.

2 In the detail band create a SubReport and select Items from the drop-down list.

3 Set the properties of the SubReport:

Left 0.5

Top 0

Width 6.3

4 In the detail band create a line component.

5 Set the line component properties:

Left 0



Top 0

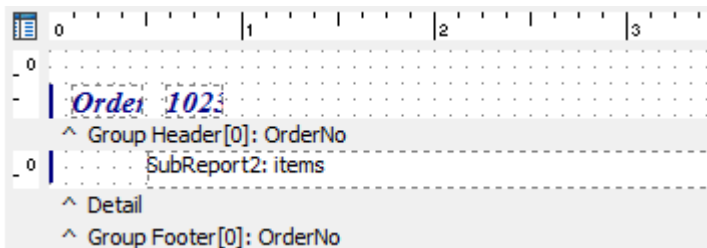
Height 0.1667

Line Color Navy Blue

Line Thickness 2 pt.

Line Position Left

6 Drag the detail band up to meet the bottom of the SubReport.



## Task 6 Layout the Title Band

1 Select the 'SubReport2: Items' tab at the bottom of the screen.

2 Create a label in the title band.

3 Set the label properties:

Top 0.0417

Font Name Times New Roman

Font Size 11

Font Style Bold and Italic

Font Color Green

4 Create 3 more labels in the title band.

5 Shift-click all the labels and align them to the top of the first label by clicking the align top icon.

6 Deselect the components and individually set the label properties from left to right:

Caption Left Alignment

ItemNo 0.2 Right

PartNo 0.95 Center

Qty 1.7 Center

Discount 2.2 Right

7 Shift-click the following labels:

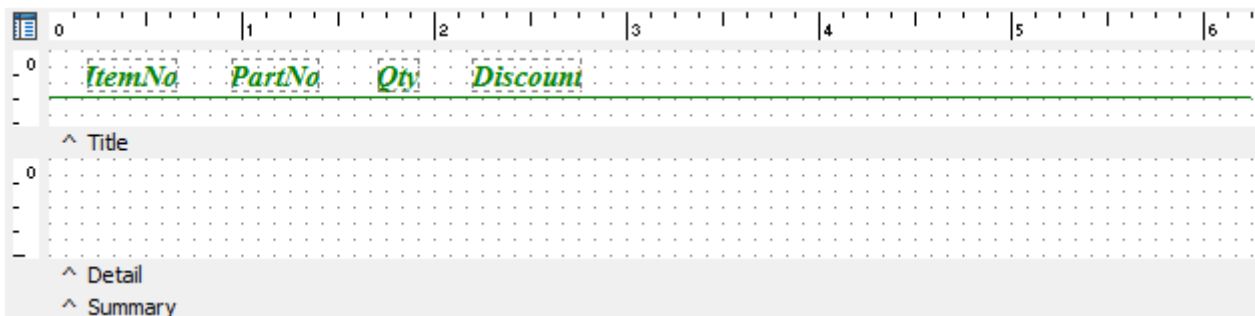
Price

Discount

- 8 Set these labels to Autosize True.
- 9 Create a line component in the title band.
- 10 Set the line component properties:

Left 0  
 Top 0.2396  
 Width 6.25  
 Line Color Green  
 Line Thickness 1 ?pt.

- 11 Drag the title band up to meet the bottom of the line component.



## Task 7 Layout the Detail Band

- 1 Create a DBText component in the detail band.
- 2 Set the DBText properties:

Top 0.0208  
 Font Name Arial  
 Font Size 8  
 Font Style Bold  
 Font Color Green

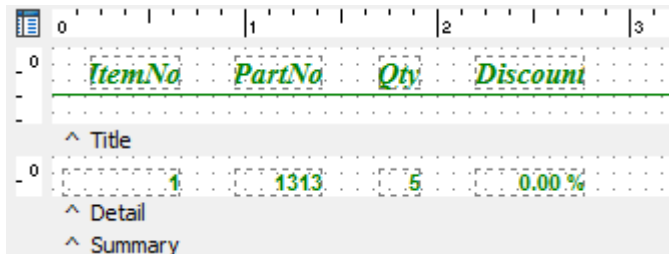
- 3 Create 3 more DBText components.
- 4 Shift-click all the components and align them to the top of the first component by clicking the align top icon.
- 5 Deselect all the DBText components and individually set the properties from left to right:

Datafield Left Alignment  
 ItemNo 0.052 Right  
 PartNo 0.8646 Center  
 Qty 1.6 Center

## Discount 2.1 Right

6 Right-click the Discount DBText component and select Display Format.

7 Choose the first format with a % sign.



ItemNo	PartNo	Qty	Discount
1	1313	5	0.00 %

^ Title  
^ Detail  
^ Summary

### Task 8 Creating a Shape

1 Place a shape in the detail band.

2 Set the shape properties:

Left 0

Top 0

Height 0.2396

Width 6.25

layout check

3 Select the shape component and click the Fill Color icon.

4 Choose More Fill Colors.

5 Click Define Custom Colors.

6 Select a custom color by entering Red, Green, and Blue values:

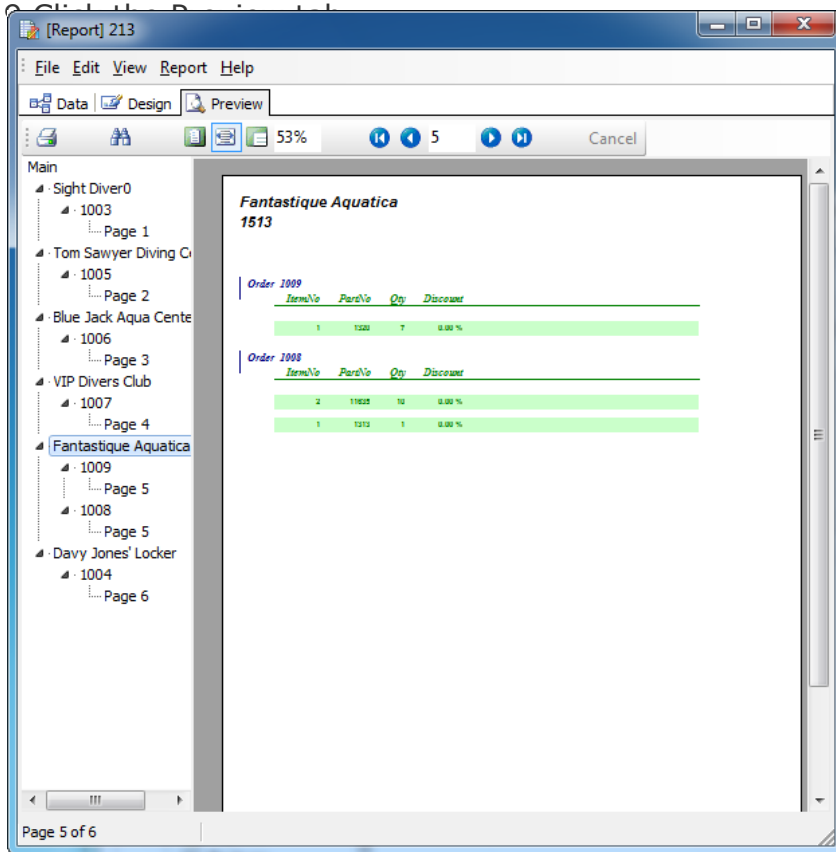
Red 202

Green 255

Blue 202

7 Drag the detail band up to meet the bottom of the shape component.

8 Right-click the shape and select Send To Back.



10 As you preview, notice how the orders for each company start on a new page, and the items for each order are listed.

11 Close the Report Designer, saving the changes. You have created a report showing data from three dataviews; a master and two details

## Sales Statistics

### OVERVIEW

In this tutorial we are going to build a report that answers some managerial questions by summarizing data from a database. The database contains sales information for an equipment wholesaler. The questions are:

- How many orders has each customer placed?
- How much product was moved with each order?
- What were the total sales per order and per customer?

We will answer these questions by selecting data at the most detailed level, and then using the report to summarize the data. The report will contain the following items:

- The customer number and name
- The orders for each customer
- The quantity and total amount for each order

### *Sales Statistics*

<i>Cust.No. Company</i>	<i>Order No.</i>	<i>Qty</i>	<i>Total</i>
1351 Slight Diver0			
	1003	5	\$1,250.00
		6	\$1,250.00
1358 Tom Sawyer Diving Centre			
	1005	21	\$18,410.55
		21	\$18,410.55
1380 Blue Jack Aqua Center			
	1008	25	\$50,036.50
		25	\$50,036.50
1384 VIP Divers Club			
	1007	20	\$8,500.00
		20	\$8,500.00
1613 Fantastique Aquatica			
	1008	11	\$1,449.50
	1009	7	\$1,197.00
		18	\$2,646.50
2158 Davy Jones' Looker			
	1004	33	\$10,535.00
		33	\$10,535.00

## ESTABLISH THE BASIC LAYOUT

### Task 1 Create the Dataview

- 1 Create a new report.
- 2 Access the data workspace.
- 3 Select File | New.
- 4 Double-click on the Query Designer icon.

- 5 Select the Customer, Orders, Items, and Parts tables.
- 6 Click the Fields tab.
- 7 Click the All Fields check box.

### **Task 3 Calculate the Total for Each Line Item**

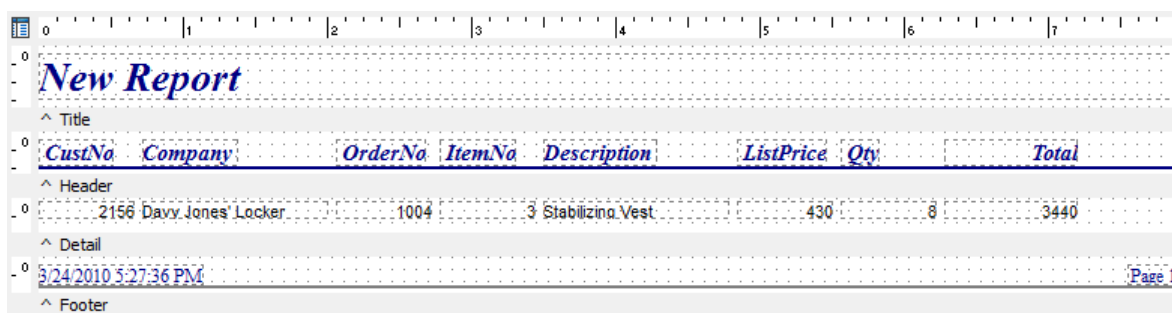
- 1 Click the Calcs tab.
- 2 Maximize the Query Designer.
- 3 Double-click on the Company field.
- 4 Select Expression from the drop-down list box in the Function column.
- 5 Enter the following calculation into the Expression edit box: Listprice \* Qty
- 6 Change the name of the Field Alias to Total.
- 7 Click on the Sort tab.
- 8 Select Custno, then Orderno.
- 9 Click OK. It may take a second or two before you see the dataview.

### **Task 4 Generate the Report via the Report Wizard**

- 1 Access the design workspace.
- 2 Hide the Data Tree if it is visible.
- 3 Select File | New.
- 4 Double-click the Report Wizard icon.
- 5 Select the following fields in order:

Custno  
Company  
Orderno  
Itemno  
Description  
Listprice  
Qty  
Total

- 6 Click Next until you reach the screen with the checkered racing flag.
- 7 Select 'Modify the report's design'.
- 8 Click Finish. The layout should look like this:



CustNo	Company	OrderNo	ItemNo	Description	ListPrice	Qty	Total
2156	Davy Jones' Locker	1004	3	Stabilizing Vest	430	8	3440

9 Press Ctrl + S and save the report under the name Sales Statistics.

10 Preview the report. This report contains data in its raw form. The data doesn't really answer the proposed questions, but it does give us something we can transform into answers. Notice the data in the Total column. We can use this total to compute the total per order and per company.

## MODIFY THE BASIC REPORT LAYOUT

### Task 1 Calculate Totals for Each Company

1 Return to the design workspace.

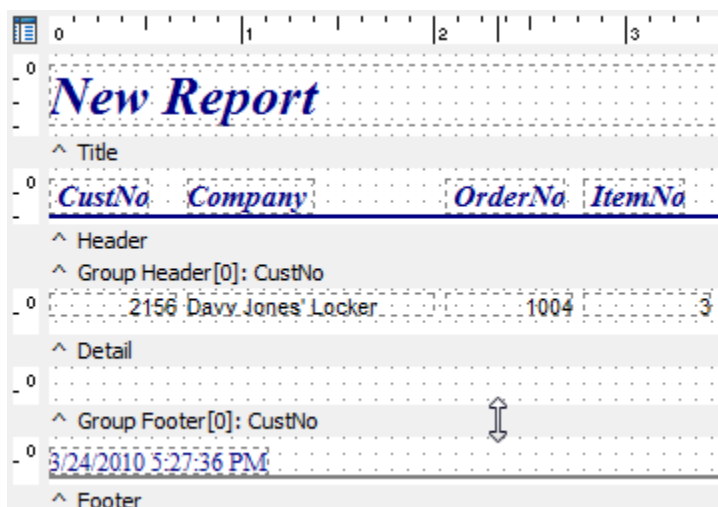
2 Select Report | Groups.

3 Click Add.

4 Assign the group to the CustNo field by expanding the drop-down list box and selecting 'CustNo'.

5 Click OK. A group header and group footer band will appear on the canvas. Notice the descriptions on these bands. The number shows which group the band belongs to and the word 'Custno' shows the field to which the group is assigned.

6 Extend the group footer band to the second tick on the vertical ruler:



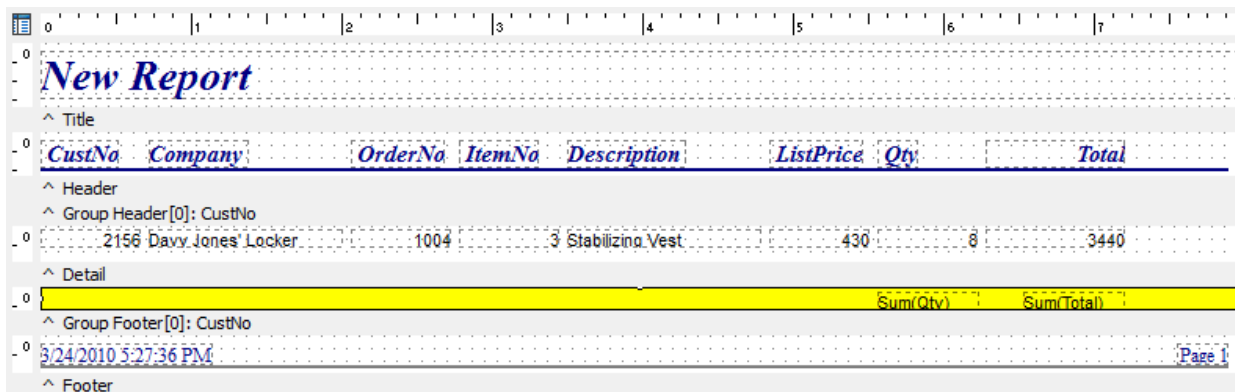
CustNo	Company	OrderNo	ItemNo
2156	Davy Jones' Locker	1004	3

7 Select a DBText component in the band. This establishes the font size and style for the DBCalc we are about to create.

- 8 Place a DBCalc component in the group footer band.
- 9 Assign it to the 'Total' field.
- 10 Right-align the DBCalc component with the Total DBText component in the detail band.
- 11 Set the top of the DBCalc component to 0.09.

## Task 2 Calculate the Total Quantity Per Company

- 1 Place another DBCalc component in the group footer band.
- 2 Assign it to the 'Qty' field.
- 3 Right-align the DBCalc with the Qty DBText component in the detail band.
- 4 Align the top of the Qty DBCalc with the Total DBCalc.
- 5 Right justify the text of the DBCalcs.
- 6 Place a shape in the group footer band and set it to ParentWidth and ParentHeight.
- 7 Send the shape to the back.
- 8 Set the shape's color to yellow. The layout should look like this:



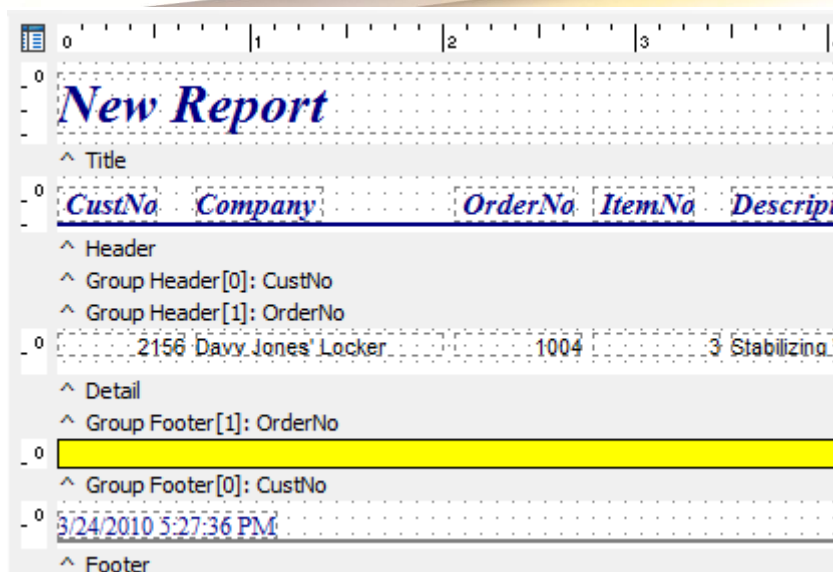
New Report							
CustNo	Company	OrderNo	ItemNo	Description	ListPrice	Qty	Total
2156	Davy Jones Locker	1004	3	Stabilizing Vest	430	8	3440
						Sum(Qty)	Sum(Total)
						Sum(Qty)	Sum(Total)
5/24/2010 5:27:36 PM Page 1							

- 9 Preview. The yellow shapes show where the group footer is printing. The group footer band contains the grand total for both the quantity of items sold and the sales revenue. Now let's calculate the total for each order.

## Task 3 Calculate Totals for Each Order

- 1 Return to the design workspace.
- 2 Select Report | Groups
- 3 Click the Add button, then select Customer. Orderno from the drop-down list box.
- 4 Click OK. New group header and group footer bands are created for the Orderno group. The layout should look like this:

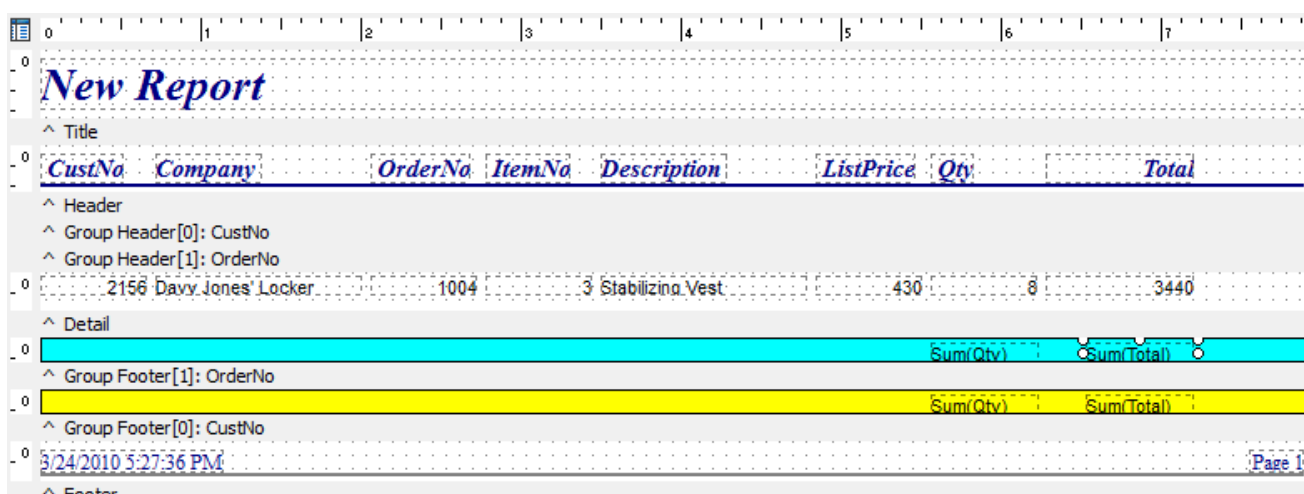




- 5 Extend the Orderno group footer to the second tick on the vertical ruler.
- 6 Copy and paste the DBCalc components in the Custno group footer band.
- 7 Drag the selection into the Orderno group footer band.
- 8 Set the top of the components to 0.09
- 9 Right-align the DBCCalc components with the corresponding DBText components in the detail band.

## Task 4 Create a Shape for the Orderno Group Footer Band

- 1 Place a shape in the band and set it to ParentHeight and ParentWidth.
- 2 Send the shape to the back.
- 3 Set the shape's color to aqua. The layout should look like this:



- 4 Preview. The aqua shapes show where the group footer is printing for each order. Notice that the customer number, company name, and order number in the detail band contain repeating values.

## **Task 5 Fix the Repeating Fields**

- 1 Return to the design workspace.
- 2 Extend the Custno group header band to the second tick on the vertical ruler.
- 3 Select the Custno and Company DBText components in the detail band and move them into the Custno group header band.
- 4 Extend the Orderno group header to the second tick on the vertical ruler.
- 5 Move the Orderno DBText from the detail band into the group header band.
- 6 Preview the report. Now the customer and order information appears only once at the top of the corresponding group.

## **FIT AND FINISH**

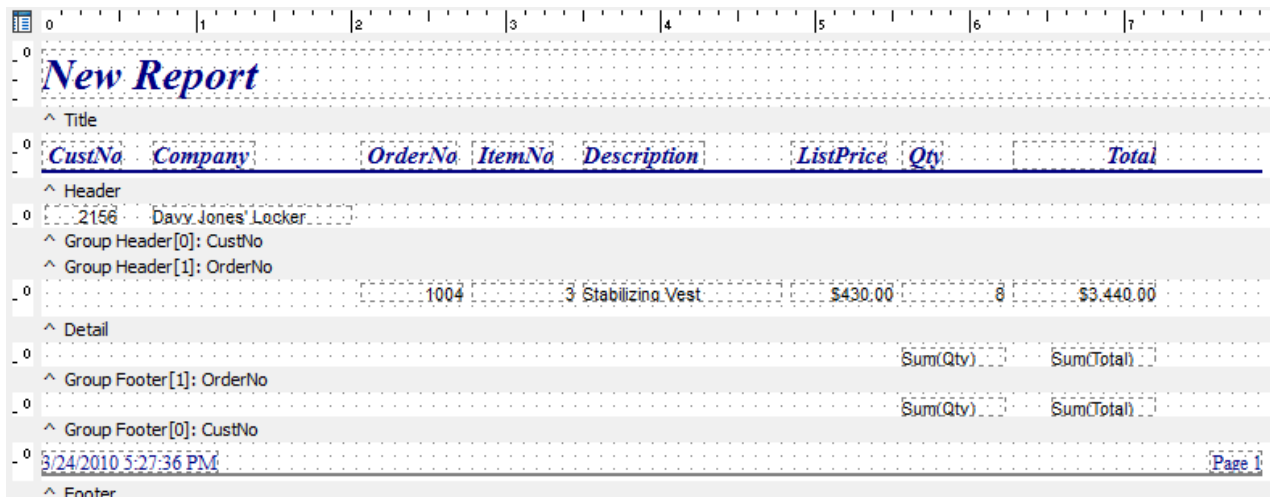
### **Task 1 Modify the Bands**

The data in the detail band is not needed to answer the questions we posed at the beginning of the tutorial. In fact, it adds unnecessary 'noise' to the report and makes it difficult to see the answers calculated in the group footer bands. We can resolve this issue by hiding the detail band.

- 1 Return to the design workspace.
- 2 Right-click over the white space of the detail band.
- 3 Select Visible, which will set this option to False.
- 4 Preview. Now only the totals are shown. However, the Orderno appears on a separate blank line. It would be easier to read the report if we placed the Orderno with the total.
- 5 Return to the design workspace.
- 6 Move the Orderno DBText from the Orderno group header band to the Orderno group footer band.
- 7 Right-align the DBText with the Orderno label in the header band.
- 8 Align the top of the DBText with the Qty

DBCalc.

- 9 Set the height of the Orderno group header band to 0.
- 10 Delete the shapes from the Orderno and Custno group footer bands. The layout should look like this:



The screenshot shows the 'New Report' window in Navicat. The report is titled 'New Report' and has a ruler at the top. The layout includes several bands: Title, Header, Group Header, Detail, Group Footer, and Footer. The data is organized into columns: CustNo, Company, OrderNo, ItemNo, Description, ListPrice, Qty, and Total. A sample record is shown with CustNo 2156, Company Davy Jones' Locker, OrderNo 1004, ItemNo 3, Description Stabilizing Vest, ListPrice \$430.00, Qty 8, and Total \$3,440.00. The report also includes summary rows for Sum(Qty) and Sum(Total) and a footer with the date 3/24/2010 5:27:36 PM and Page 1.

CustNo	Company	OrderNo	ItemNo	Description	ListPrice	Qty	Total
2156	Davy Jones' Locker	1004	3	Stabilizing Vest	\$430.00	8	\$3,440.00
						Sum(Qty)	Sum(Total)

Note: The shapes were used to help us quickly identify where the group footer bands were print-ing. Now that we have hidden the detail band, they are no longer necessary.

## Task 2 Set the Format of the DBCalcs

- 1 Right-click over the Total DBCalc in the Custno group footer band and select Display Format.
- 2 Select the first menu option with a dollar sign.
- 3 Right-click over the Total DBCalc in the Orderno group footer band and select Display For-mat.
- 4 Select the first menu option with a dollar sign.

## Task 3 Complete the Custno Group Footer Band

- 1 Select the DBCalc components in the Custno group footer band and set the font to bold.
- 2 Place a shape in the Custno group footer band.
- 3 Set the following positions for the shape:

Left 0.05

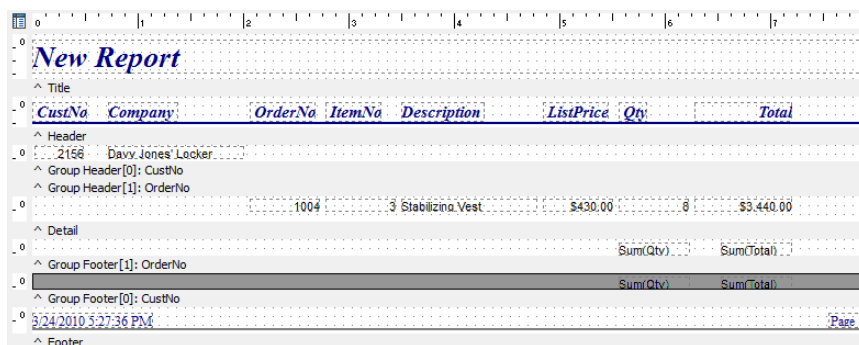
Top 0. 06

Width 7.95

Height 0.18

- 4 Send the shape to the back.

5 Set the color of the shape to light gray. The layout should look like this:



New Report							
CustNo	Company	OrderNo	ItemNo	Description	ListPrice	Qty	Total
2156	Davy Jones' Locker	1004	3	Stabilizing Vest	\$430.00	8	\$3,440.00
Sum(Qty)					Sum(Total)		
Sum(Qty)					Sum(Total)		
Sum(Qty)					Sum(Total)		
3/24/2010 5:27:36 PM Page 1							

## Task 4 Put the Finishing Touches on the Report

- 1 Select the Custno and Company DBTexts in the Custno group header band and the Orderno DBText in the Orderno group footer band.
- 2 Set the font color to navy.
- 3 Set the font to bold.
- 4 Set the Visible speed menu option to False for the following labels:

Itemno

Description

Listprice

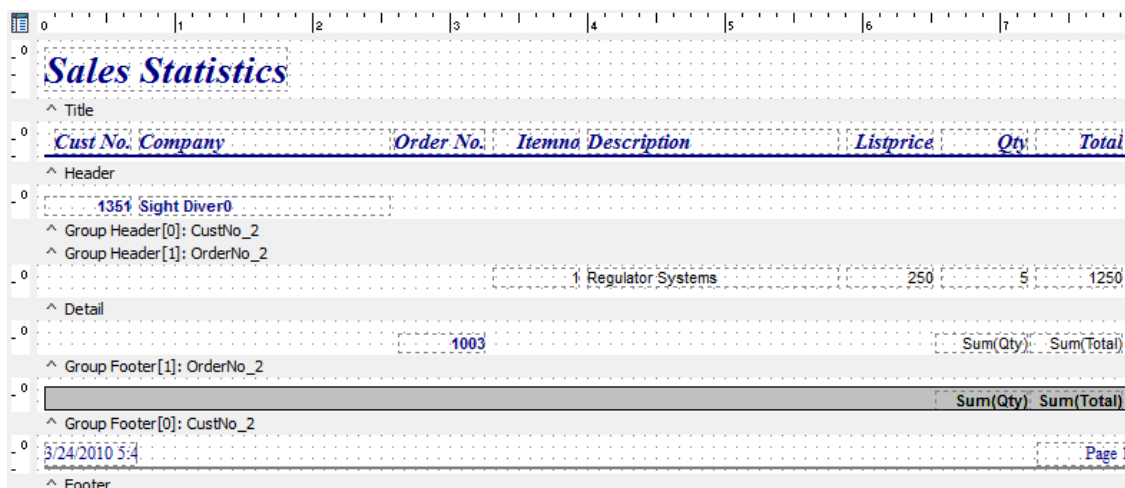
- 5 Change the text for these labels accordingly:

New Report Sales Statistics

Custno Customer No.

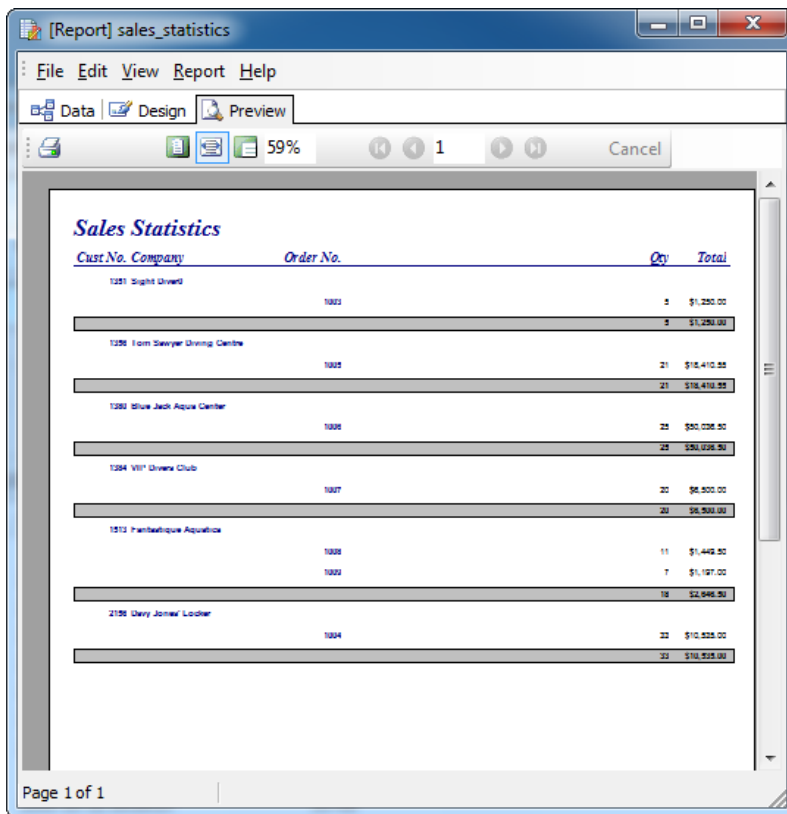
Orderno Order No.

- 6 Autosize these three labels. The final layout should look like this:



Sales Statistics							
Cust No.	Company	Order No.	Itemno	Description	Listprice	Qty	Total
1351	Sight Diver0	1	1	Regulator Systems	250	5	1250
Sum(Qty)					Sum(Total)		
Sum(Qty)					Sum(Total)		
Sum(Qty)					Sum(Total)		
3/24/2010 5:4 Page 1							

7 Preview. The final report should look this:



**Sales Statistics**

<u>Cust.No.</u>	<u>Company</u>	<u>Order No.</u>	<u>Qty</u>	<u>Total</u>
1281	Sight United	1002	5	\$1,250.00
			5	\$1,250.00
1286	Tom Sawyer Driving Centre	1003	21	\$18,410.55
			21	\$18,410.55
1282	Blue Jack Aquatic Center	1006	25	\$50,036.50
			25	\$50,036.50
1284	Vitt' Drivers Club	1007	20	\$6,500.00
			20	\$6,500.00
1513	Paradise Aquatics	1008	11	\$1,442.50
		1009	7	\$1,127.00
			18	\$2,569.50
2190	Garry Jones' Locker	1004	33	\$10,525.00
			33	\$10,525.00

Page 1 of 1

8 Close the Report Designer, saving the changes.

## Crosstab

### OVERVIEW

This tutorial shows how to create a report using the Crosstab tool. In order to fully understand the concept of crosstabs, we need to create physical printouts and look at them in different ways; therefore, you'll need access to a printer, some scissors, tape, and a stapler for this tutorial. By the end of this tutorial we will have created the following products:

- A poster-style crosstab using 'down then across' pagination
- A poster-style crosstab using 'across then down' pagination
- A document-style crosstab using the standard style
- A document-style crosstab using the repeated captions style


### Task 1 Select Data

- 1 Create a new report.
- 2 Hide the Data Tree and the Report Tree if they are visible.
- 3 Access the Query Designer.
- 4 Select the Customer and Orders tables.
- 5 Click the Fields tab and select All Fields.
- 6 Scroll down to the Amount Paid field and select it, wait a second, then click again.
- 7 Change the name to Sale Amount.
- 8 Click the Calcs tab and double-click on the Company field.
- 9 Maximize the Query Designer, then choose Expression from the Function drop-down list and enter the following expression:  
Extract (Month from SaleDate)

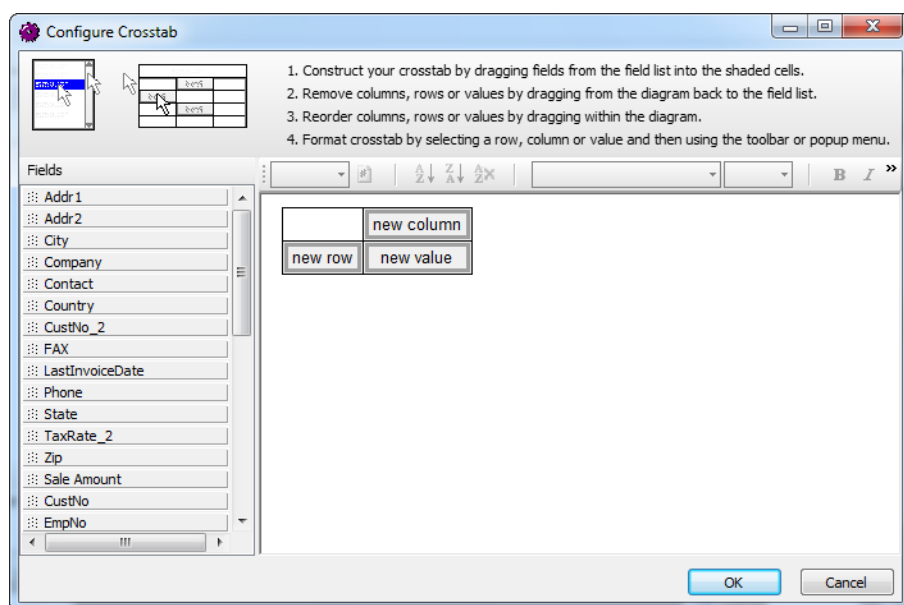
Note: This expression takes the month from the sale date, allowing us to compute totals on a monthly basis.

- 10 Change the Field Alias to SaleMonth.
- 11 Click OK.

### Task 2 Create a Crosstab

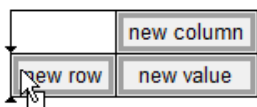
- 1 Access the design workspace.
- 2 Select Report | Data. Select None for both Pipelines. Click OK.
- 3 Place a crosstab component  in the detail band.
- 4 Use the Edit toolbar to assign the crosstab to the Customer pipeline
- 5 Select File | Page Setup. Access the Paper Size tab and set the orientation to Landscape. Click OK.

6 Right-click over the crosstab and select Configure. The Crosstab Designer will be displayed:



7 Read the instructions at the top of the Crosstab Designer.

8 Select the SaleMonth field (at the bottom of the list) and drag it over the new row cell. Look for the black, triangular indicators that show where the dimension will be created:



9 When the indicators appear to the left of the new row element, drop the field into the diagram. The diagram should look like this:

		new column
SaleMonth	new row	new value

10 Click OK.

11 Press Ctrl + S and save the report under the name Crosstab.

12 Preview. A blank page is displayed because no values have been assigned to the crosstab.

### Task 3 Design the Crosstab

1 Return to the design workspace, right-click over the crosstab, and select Configure.

2 Drag the Sale Amount field over the new value cell. When the indicators appear, drop the field into the diagram. The diagram should look like this:

		new column
SaleMonth	new row	Sum of Sale Amount 1000.00
		new value
Grand Total		1000.00

Note: The number 1000 represents the format of the calculated value. The Grand Total indicates that the last row of the crosstab will show the total sale amount for all months.

3 Select SaleMonth and use the highlight color palette to set the color to green. Notice that the element turns fuchsia instead of green. This is because the cell is selected. Deselect it by clicking another element and it should turn green.

4 Select Grand Total and set the color to yellow.

5 Select Sum of Sale Amount and set the color to fuchsia.

6 Select the 1000 under Sum of Sale Amount and set the color to red.

7 Select the 1000 to the right of the grand total and set the color to gray.

8 Close the Crosstab Designer.

9 Preview. The colors allow you to see where each element prints. The values in red represent the sale amount per month. The value in gray is the grand total for the year. The numbers in green represent the months. The yellow and fuchsia sections show where the headings, or captions, print. Now let's add some more values. Right now we have the sum of sales. Next we'll add the average and number of sales per month (count) to the crosstab.

## EXTEND THE CROSSTAB DESIGN

### Task 1 Add Values to the Crosstab

1 Access the Crosstab Designer.

2 Select all colored sections and set the color to none.

3 Drag the Sale Amount field over the new value cell and release it.

4 Select the second Sum of Sale Amount (after the new value cell).

5 Locate the drop-down list box on the toolbar.

6 Select Average from the drop-down list.

7 Once again, drag the Sale Amount field over to the new value cell and release it.

8 Select the second Sum of Sale Amount (the one below the average).

9 Select Count from the drop-down list. The diagram should look like this:



		new column
SaleMonth	new row	Sum of Sale Amount 1000.00
		Average Sale Amount 1000.00
		Count of Sale Amount 1000
		new value
Grand Total (Sum of Sale Amount)		1000.00
Grand Total (Average Sale Amount)		1000.00
Grand Total (Count of Sale Amount)		1000

10 Click OK.

11 Preview. The crosstab includes new values. Advance to the second page. The grand totals are on this page. The report tells us the sum, average, and count for the sale amount per month.

## Task 2 Set the Format of the Values

- 1 Access the Crosstab Designer.
- 2 Select the 1000 under Sum of Sale Amount.
- 3 Right-click and select Display Format. Select the first menu option with a dollar sign.
- 4 Select the 1000 under Average Sale Amount.
- 5 Right-click and select Display Format. Select the first menu option with a dollar sign.

## Task 3 Calculate Totals by State

- 1 Drag the SaleMonth cell over the new column cell and release.

		new column
SaleMonth	new row	Sum of Sale Amount \$1,000.00
		Average Sale Amount \$1,000.00
		Count of Sale Amount 1000
		new value
Grand Total (Sum of Sale Amount)		\$1,000.00
Grand Total (Average Sale Amount)		\$1,000.00
Grand Total (Count of Sale Amount)		1000

2 Drag the State field over the new row cell and release. The diagram should look like this:

		SaleMonth	Grand Total
		new column	
State	new row	Sum of Sale Amount \$1,000.00	\$1,000.00
		Average Sale Amount \$1,000.00	\$1,000.00
		Count of Sale Amount 1000	1000
		new value	
Grand Total (Sum of Sale Amount)		\$1,000.00	
Grand Total (Average Sale Amount)		\$1,000.00	
Grand Total (Count of Sale Amount)		1000	

3 Click OK.

4 Preview. Select Whole Page on the preview toolbar. The crosstab is reformatted. Notice that the months go across the top of the report instead of down the side. This is because we made SaleMonth a column. The crosstab shows us the sum, average, and count for the sale amount per month for each state. Unfortunately, the preview screen doesn't give us a complete view of the entire crosstab. After we lay out the header band, we'll print out and assemble a complete view of the crosstab.

## Task 4 Lay Out the Header Band

1 Return to the design workspace and place a label in the upper left corner of the header band.

2 Set the caption to Annual Sales.

3 Set the font to 12 bold.

4 Place a System Variable component in the lower left corner of the header band.

5 Set it to PrintDateTime.

6 Place another System Variable in the lower right corner of the header band.

7 Set it to PageSetDesc.

8 Align the tops of the system variables.

9 Select Ctrl + S to save the report.

10 Preview. Notice that the page number prints on each page.

## UNDERSTANDING CROSSTAB LAYOUTS

### Task 1 Control Pagination: Down then Across

- 1 From the Preview screen, print out all four pages of the crosstab.
- 2 Return to the design workspace and right click over the crosstab.
- 3 Select Pagination. Notice that the pagination defaults to Down then Across. This setting refers to the order in which the pages print.
- 4 Retrieve your printed pages and lay them out like this:

1	3
2	4

- 5 Cut off the 1/2 inch bottom margin of pages one and three. Cut off the 1/4 inch left margin of pages three and four.

6 Tape the pages together, using the crosstab grid to align them properly:

1	3
2	4

- 7 Label this crosstab 'Down then Across'.

### Task 2 Control Pagination: Across then Down

- 1 Right-click over the crosstab component and select Pagination.
- 2 Click on Across then Down.
- 3 Preview.
- 4 Print all four pages of the crosstab.
- 5 Retrieve your printed pages and lay them out like this:

1	2
3	4

- 6 Cut off the 1/2 inch bottom margin of pages one and two. Cut off the 1/4 inch left margin of pages two and four.

7 Tape the pages together, using the crosstab grid to align them properly:

1	2
3	4

- 8 Label this crosstab 'Across then Down.'

Note: As you can see, pagination controls the order in which the pages print when the crosstab cannot fit on a single page.

### **Task 3 Use Repeated Captions**

- 1 Click the Design tab.
- 2 Right-click over the crosstab component.
- 3 Expand the Style menu option. Notice that the default setting is Standard. This means that the captions do not repeat. Both of the crosstabs we put together are set to Standard because we did not want to see the captions on every page. However, if we print the crosstab as a document, we may want the captions to repeat for clarity.
- 4 Select Repeated Captions.
- 5 Preview and print all four pages of the crosstab.
- 6 Put them in order by page number and staple the pages together.
- 7 Flip through them. Notice that the captions for the state and the month appear on each page.
- 8 Right-click over the crosstab and set the Style to Standard.
- 9 Preview and print.
- 10 Staple the pages together.
- 11 Flip through them. Notice that the captions from page one do not repeat on page two. When we print a crosstab as a document, we can use repeated captions to keep track of values that appear on subsequent pages.
- 12 Close the Report Designer and save the changes. Congratulations, you've completed all of the tutorials.

## Mailing Label

### OVERVIEW

The first section of this tutorial demonstrates how to print mailing labels via the Label Template Wizard, which is a tool that creates report layouts suitable for printing on label sheets. Each label will include the following items:

- A contact name
- The company name for each contact
- The mailing address for each contact

Charles Lopez Divers of Cortu, Inc. Marmoset Place 54 Aylos Matthalos, Cortu	Anna Rack Safari Under the Sea PO Box 7456 Grand Cayman,	Marianne Miles Action Diver Supply Blue Spar Box #3 St. Thomas, 00820
Anne Mariachi Princess Island SCUBA PO Box 32 Waiyevo Taveuni,	Kevin Rider Tora Tora Tora PO Box H-4573 Nassau,	Russell Christopher VIP Divers Club 32 Main St. Christiansted, St. Croix 02800
Grant Alsworth Underwater Fantasy PO Box 842 Ocho Rios, Jamaica	George Weathers Uniscop1 PO Box Z-547 Freeport,	Maria Eventosh Central Underwater Supplies PO Box 737 Johannesburg, 2042
Barbara Harvey Jamaica SCUBA Centre PO Box 68 Negril, Jamaica	Phyllis Spooner Sight Diver0 1 Neptune Lane Kato Paphos,	Bill Wyers George Bean & Co. #73 King Salmon Way Lugoff, NC 29078

### THE LABEL TEMPLATE WIZARD

#### Task 1 Select Data

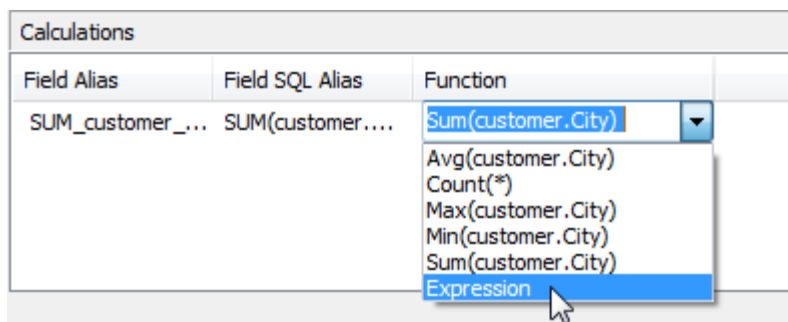
- 1 Create a new report.
- 2 Access the data workspace.
- 3 Select File | New from the main menu.
- 4 Double-click on the Query Designer icon:
- 5 Maximize the Query Designer by clicking the Maximize icon in the title bar:
- 6 Double-click the Customer table.
- 7 Click the Fields tab.
- 8 Click the All Fields check box:

#### Task 2 Create a Calculated Field

- 1 Click the Calcs tab.
- 2 Double-click the City field. SUM\_customer\_City will appear as the first calculation in the list at the bottom half of the screen:

3 Click on the new SUM\_customer\_City calculation.

4 Expand the drop-down list in the Function column and select Expression from the list. An edit box will appear in the Expression column.



5 Place your cursor in the edit box.

6 Enter the following expression:

`concat(city, ' ', state, ' ', zip)`

Each element of this expression has the following meaning:

**concat(city, ' ', state, ' ', zip)**

A	B	C	D	E

`concat(city, ' ', state, ' ', zip)`

A Gets the value of the City field

B Places a comma and a space between the city and state

C Gets the value of the State field

D Places a space between the state and the zip

E Gets the value of the Zip field

The concat represents the concatenation function. Concatenation is a process by which values are combined into one result. A typical result for this expression would be 'Columbia, MO 65203'.

A B C D E

7 Rename the SUM\_customer\_City field alias by clicking once on SUM\_customer\_City, then waiting a second. This will put the field alias in 'edit' mode. Type CityStateZip in the edit box. The Calculation field should look like this:

### Task 3 Sort the Data

1 Click the Sort tab.

2 Double-click on Zip.

3 Click OK. A dataview will appear. Click the preview icon on the dataview:

4 Scroll to the right until you see the Zip field. Notice that the data is ordered by ZIP Code (the companies with no ZIP Code appear first.). The last column should contain the combination of the city, state, and ZIP Code, as specified in the calculated field we created.

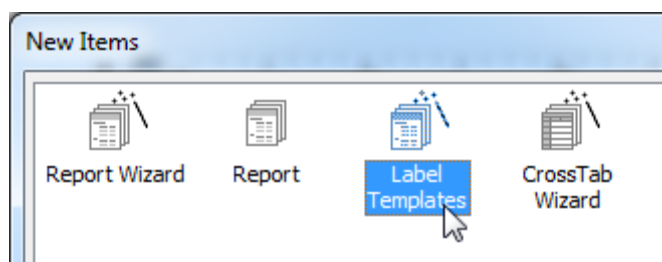
5 Click OK.

## Task 4 Configure the Report Layout

1 Access the design workspace.

2 Select File | New from the main menu.

3 Double-click on the Label Templates icon:



4 Make the following selections:

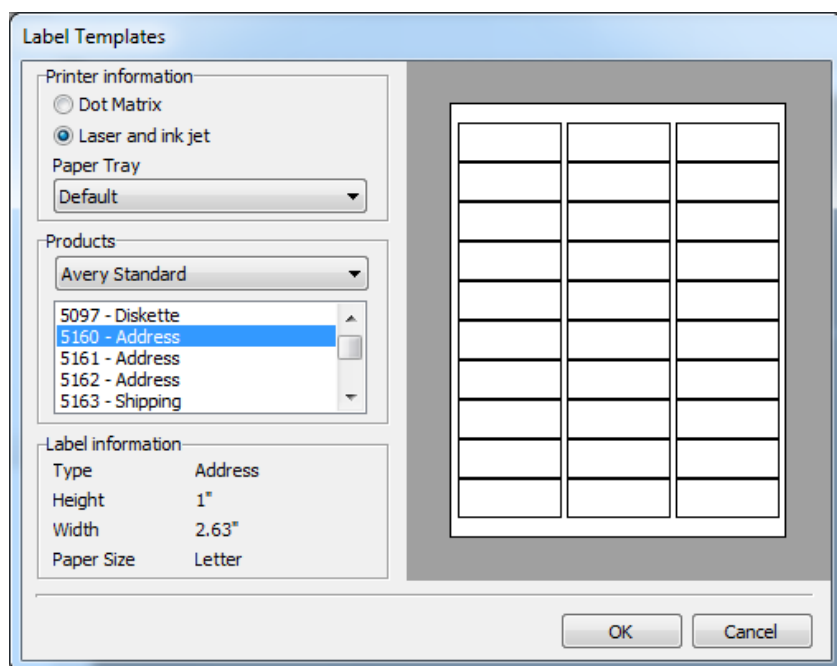
Printer information Laser and ink jet

Paper Tray Manual Feed (Tray 1)

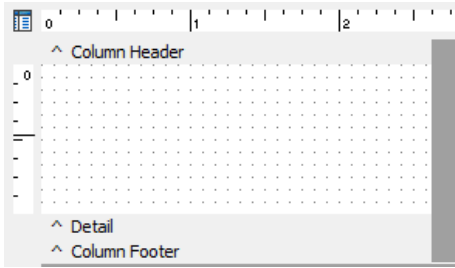
Products Avery Standard

5160 – Address

The Label Template dialog box should look like this:

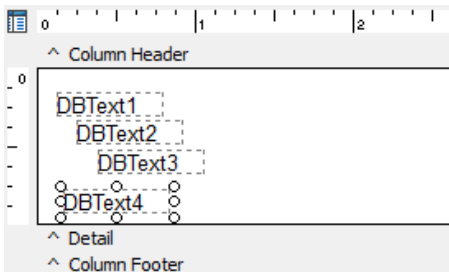


5 Select OK. The report layout should look like this:



## Task 5 Lay Out the Detail Band

- 1 Place a shape in the detail band.
- 2 Set the shape to ParentHeight and ParentWidth.
- 3 Preview the report. The shape indicates where the detail band is printing. Notice how the detail band prints in a columnar pattern and is positioned just like the labels on the sheet.
- 4 Place four DBText components in the detail band:



- 5 Set the following positions for the DBText components:

DBText1 Left 0.08  
 Top 0.06  
 DBText 4 Left 0.08  
 Top 0.77

- 6 Select DBText1, then shift-click the other components.
- 7 Align the components left and space them vertically.
- 8 Right-click over each component and set it to AutoSize.
- 9 Assign the components to the following fields:

DBText1 Contact  
 DBText2 Company  
 DBText3 Addr1  
 DBText4 CityStateZip

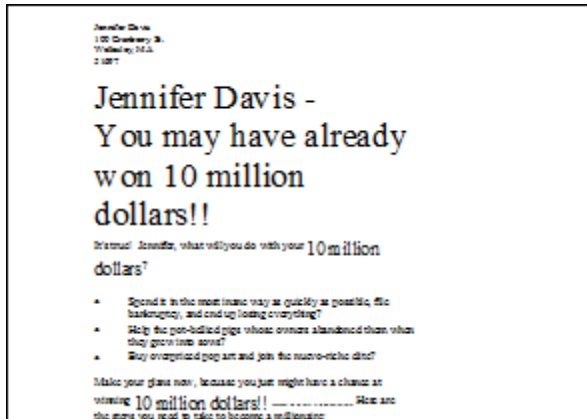


## Form Letter

### OVERVIEW

This tutorial illustrates how to create a form letter. The report will include these items:

- A letter for each customer in the database
- The name and return address of the recipient within each letter



### Task 1 Create a Brief Letter

The first thing you'll need to do is compose a 23 line letter in WordPad. We'll use this letter to illustrate the process of opening a document via the RichText component and changing it into a Form Letter.

1 Select Start | Programs | Accessories | WordPad.

2 Compose a 2-3 line letter addressed to:

Earl Gray

1000 Pocahontas Drive

Burlington, KY 45001

3 Select File | Save from the WordPad main menu.

4 Select Rich Text Format from the 'Save as Type' drop-down list box.

5 Locate the directory

6 Save the file under the name My Letter.

7 Shut down WordPad.

### BUILD THE REPORT


#### Task 1 Select Data

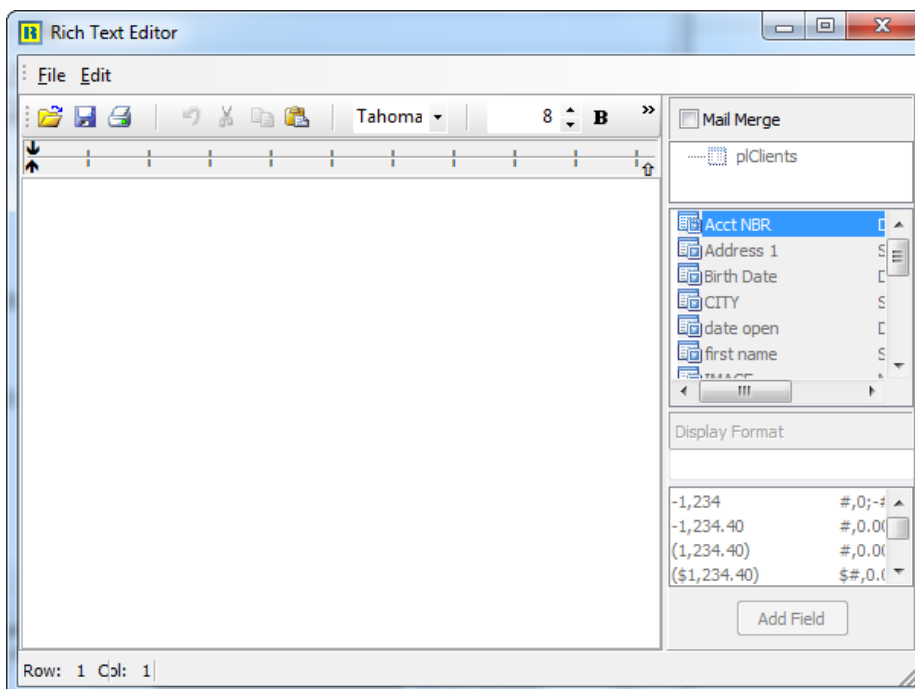
1 Return to ReportBuilder.

2 Create a new report.

- 3 Access the Query Wizard.
- 4 Select the Clients table.
- 5 Click Finish.

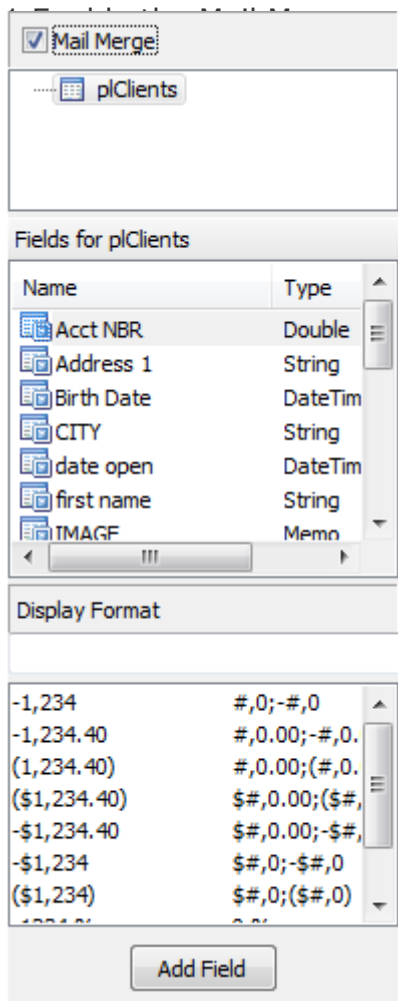
## Task 2 Lay Out the Detail Band

- 1 Access the design workspace.
- 2 Select the Report menu option and remove the header and footer bands.
- 3 Expand the detail band to one inch.
- 4 Place a RichText  component in the detail band.
- 5 Align the left edge of the component with the one inch mark on the horizontal ruler.
- 6 Right-click over the component and select MailMerge.
- 7 Right-click and select Edit. A new window will open:



- 8 Select File | Open. Locate your letter and click Open.
- 9 Your letter should appear. Maximize the win-dow.

## Task 3 Make My Letter into a Form Letter



- 2 Double-click on FIRST\_NAME to insert the first name.
- 3 Double-click on LAST\_NAME to insert the last name.
- 4 Close the window and click Yes when the save prompt appears.

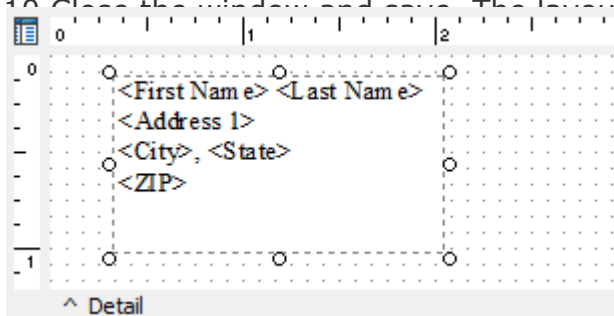
## Task 4 Experiment with MailMerge

- 1 Preview. The letter is now addressed to the people whose names are in the database.
- 2 Return to the design workspace.
- 3 Right-click over the RichText component and click MailMerge to deselect it.
- 4 Right-click and select Edit.
- 5 The Field option is not available. The MailMerge option allows you to insert fields, changing static text into form letters; therefore, if MailMerge is not selected, you cannot insert fields into a letter.
- 6 Close the window.

## Task 5 Create a Form Letter

- 1 Return to the design workspace.
- 2 Right-click over the Rich Text component and select the MailMerge option.
- 3 Right-click and select Edit.
- 4 Open 'Form Letter.rtf' in the directory containing your ReportBuilder application. This is a mock form letter. We want to be able to send it to all the people in the client table; therefore, we're going to need to replace the current name and return address with fields that contain data for each person.
- 5 Double-click the name Jane in the return address at the upper left corner of the screen.
- 6 Insert the First Name field. The name Jane should be replaced with the First Name field.
- 7 Repeat the process for the following fields:  
last name  
address  
city  
state  
zip

- 8 Replace Jane and Plain in the first line of the letter with the first and last name fields.
- 9 Replace Jane in line nine with the first name field.
- 10 Close the window and save. The layout should look like this:



- 11 Press Ctrl + S and save the report under the name Form Letter.
- 12 Preview. The name and return address for the each person in the database appear, but the letter does not. In order for the letter to print, we must change some settings on the Rich Text component.

## Task 6 Modify the Component

- 1 Return to the design workspace.
- 2 Right-click the component and select Stretch.
- 3 Preview. A portion of the letter prints, but the text is truncated on the right due to the width of the component.
- 4 Return to the design workspace.

5 Select the component and widen it by dragging the right side to the 6.5 mark on the horizontal ruler.

6 Preview. The first letter is addressed to Jennifer Davis. Scroll down. Notice that the second letter (addressed to Arthur Jones) prints at the bottom of this page. We want each letter to print on a separate page.

## Task 7 Create a Group

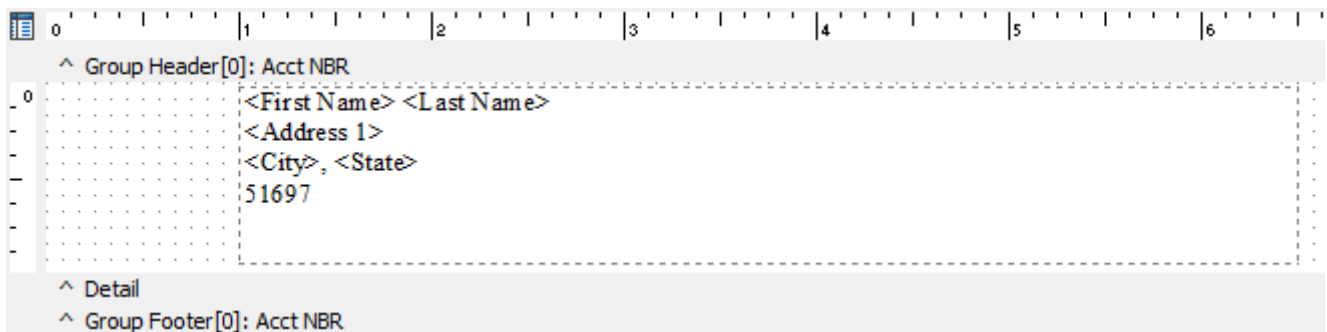
1 Return to the design workspace.

2 Select Report | Groups.

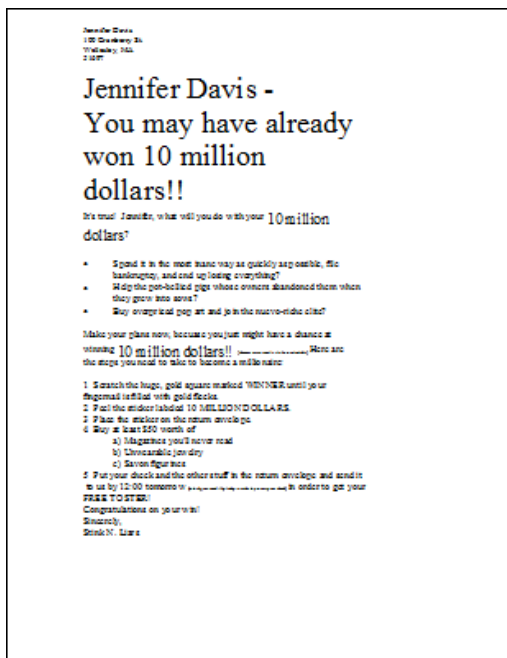
3 Click Add and select Clients.Acct.Nbr from the drop-down list box.

4 Check Start New Page.

5 Click OK. The layout should look like this:



6 Preview. Now when there is a new account number, a new group begins; therefore, the problem of a letter printing immediately after an existing one is resolved. Here is what the final preview screen should look like:



7 Close the Report Designer and save the changes.


## Photo Album

## OVERVIEW

This tutorial introduces you to the handling of images and text. The final report will contain the following items:

- A picture of a fish
- A description of the fish, which is a few paragraphs in length
- Additional information about the fish

(length, species, and category)

<b>Clown Triggerfish</b> 	<b>Triggertfish</b>
Category: Triggerfish Length In: 10.662099700767 Species Name: <i>Pomacentrus coelestis</i>	

## BUILD THE REPORT LAYOUT

## Task 1 Select Data via the Query Wizard

- 1 Create a new report.
- 2 Access the Query Wizard.
- 3 Select the Biolife table.
- 4 Click Next until you reach the screen with the Set Order option.
- 5 Choose Select Group.
- 6 Choose Set Order option.
- 7 Click Finish.

## Task 2 Lay Out the Detail Band

- 1 Access the design workspace.
- 2 Extend the detail band to two inches.
- 3 Place a shape in the detail band.
- 4 Set the fill color of the shape to red.
- 5 Set the size and position:  
Left 0.45  
Top 0  
Width 2.7  
Height 1.1
- 6 Place a DBText component over the shape.
- 7 Set the size and position:


Left 0.5  
Top 0. 1  
Width 2.6

8 Assign the Common Name field to the DBText component by selecting 'Common Name' from the drop-down list box at the upper left corner of the Report Designer.

9 Set the font size to 20 pt.

10 Set the font color to white.

### Task 3 Create a Graphic

1 Place a DBImage component  over the shape, just below the DBText component.

2 Set the position to:

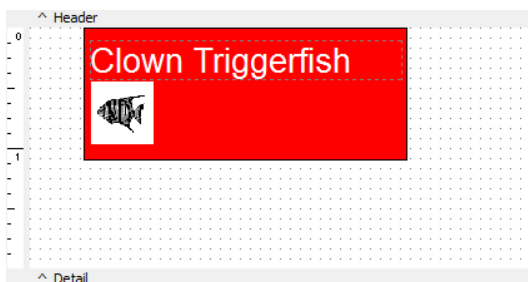
Left 0.5

Top 0.5

3 Assign the 'Graphic' field to the DBImage component. Notice that the image is cropped because it is too big for the component.

4 Right-click and select Stretch. The entire image of the fish should now be visible. This image is slightly distorted because it natively has a rectangular shape and the Stretch property is scaling it to fit into a square.

5 Right-click and select MaintainAspectRatio. The image should be readjust so that it is no longer distorted. While the image no longer fills the entire DBImage component, it looks better because the height and width of the original image have been scaled by the same amount. Aspect ratio is the ratio of the height to the width. Maintaining the aspect ratio is simply scaling the height and the width by the same percentage.



### Task 4 Use the Data Tree to Create Components

1 Create another shape and place it below the red shape in the detail band.

2 Set the size and position:

Left 0.45

Top 1. 0

Width 2.7  
Height 0.8

3 Launch the Data Tree. If it does not appear docked on the left side of the Report Designer, dock it there.

4 Select the Layout tab at the bottom of the Data Tree and set the Style to Vertical.

5 Select the Data tab at the bottom of the Data Tree and Ctrl-click the following fields:

Category

Length In

Species Name

6 Drag the selection to the upper left corner of the new shape and release the mouse button. The canvas should look like this:



7 While all of the components are still selected, change the font size to 8 pt.

8 Autosize the labels (the components on the left).

9 Set the left of each label to 0.6.

10 Set the size and position of the DBText components:

Left 1.5

Width 1.6

11 Select the labels and DBText components. Hold down the Ctrl key and use the arrow keys to move the selection up and down. Notice the Top value in the status bar at the bottom right of the Report Designer. This value is updated each time you move the selection. Use the arrow keys to move the selection to 1.1042. The layout should look like this:





## Task 5 Generate a Framing Shape and Text

1 Create another shape and place it to the right of the red shape.


2 Set the size and position:

Left 3.17

Top 0. 01

Width 4.15

Height 1.8

3 Place a DBMemo component  over the shape.

4 Set the size and position:

Left 3.25

Top 0. 08

Width 4

Height 1.65.

5 Assign the DBMemo component to the 'Category' field.

6 Set the DBMemo font:

Color red

Size 12 pt

Style bold

7 Press Ctrl + S and save the report under the name Fish Photo.

8 Preview the report.

## FIT AND FINISH

### Task 1 Adjust the Text and Shape

As you preview, notice that the description for the 'Atlantic Spadefish' is not shown completely. Let's fix this first.

1 Return to the design workspace.

2 Right-click over the DBMemo and select Stretch.

3 Preview. Now the entire description appears, but the shape bordering it does not stretch properly:



<b>Clown Triggerfish</b>  Category: Triggerfish Length In: 19.660000000000007 Species Name: Pteropoda coccinellum	Triggerfish
<b>Red Emperor</b>  Category: Snapper Length In: 22.620000000000002 Species Name: Lutjanus sebae	Snapper

- 4 Return to the design workspace and right-click over the shape behind the DBMemo
- 5 Select StretchWithParent.
- 6 Preview. The shape now functions as a frame around the description.
- 7 Scroll to the bottom of page one. Notice that the last description does not print completely on this page. Go to the next page and scroll to the top. The description finishes printing at the top of this page, but there is no framing shape.
- 8 Return to the design workspace and right-click over the shape.
- 9 Select ReprintOnOverflow. Preview and advance to page two. The shape now appears on this page as well. Next, we want to extend the shape with the DBText components so that it matches the height of the framing shape.
- 10 Return to the design workspace.
- 11 Right-click over the shape with the DBText components and select StretchWithParent, then Preview. Now we have a great-looking page.

## Task 2 Improve the Pagination

Although we have modified this layout to look as professional as possible when the description overflows onto an additional page, our preference is to keep the memo from breaking across pages in the first place. We can achieve this by using a group to hold the detail band together.

- 1 Return to the design workspace.
- 2 Select Report | Groups.
- 3 Select Biolife.CommonName.
- 4 Click the Add button. A group is created on the Common Name field. Notice that the 'Keep group together' box is checked.
- 5 Click OK.
- 6 Preview. Advance to page two. The California Moray entry is now printed completely on this page. Page two of the final report should look like this:

<div>Clown Triggerfish</div> <div></div> <div><div>CategoryTriggerfish</div><div>Length In19.6620982700747</div><div>Species NamePomacentrus coelestis</div></div>	<div>Triggerfish</div>
<div>Red Emperor</div> <div></div> <div><div>CategorySnapper</div><div>Length In22.62200472440942</div><div>Species NameLutjanus sebae</div></div>	<div>Snapper</div>

Note: Here we created a group that breaks for every record in the report. The group has the Keep Together option set to True, which means that if the group header, detail, and group footer bands cannot fit on the current page, then the group will break to the next page. Thus, when the description in the detail band didn't fit on page one, the group moved it to page two.

7 Close the Report Designer, saving the changes.


## Server Security Management

Navicat provides server security management for MySQL, Oracle, PostgreSQL and SQL Server.

- [MySQL Security Management](#)
- [Oracle Security Management](#)
- [PostgreSQL Security Management](#)
- [SQL Server Security Management](#)
- [Privilege Manager](#)




## MySQL Security Management

Navicat provides **User** to add, duplicate, edit, delete users, grant/revoke server privileges and privileges on the selected databases, tables/views, fields and functions/procedures. The object pane displays all the users that exist in the **user** table.

Just simply click  to open an object pane for **User**. A right-click displays the popup menu or use the object pane toolbar, allowing you to add, edit and delete users.


### Add User

To add a new user

- Select the connection you wish to set privileges in the navigation pane.
- Click  to open the **User** showing the user list.
- Click the  **New User** from the object pane toolbar or right-click and select  **New User** from the popup menu.
- Edit user properties and privileges on the appropriate tabs of the User Designer.




### Duplicate User

To create a new user with modification as one of the existing users

- Select the connection you wish to set privileges in the navigation pane.
- Click  to open the **User** showing the user list.
- Select a user to edit in the object pane.
- Right-click the user and select **Duplicate User** from the popup menu.
- Edit user properties and privileges on the appropriate tabs of the User Designer.




### Edit User

To edit an existing user

- Select the connection you wish to set privileges in the navigation pane.
- Click  to open the **User** showing the user list.
- Select a user to edit in the object pane.
- Click the  **Edit User** from the object pane toolbar or right-click the user and select  **Edit User** from the popup menu.
- Edit user properties and privileges on the appropriate tabs of the User Designer.



## Delete User

To delete a user

- Select the connection you wish to set privileges in the navigation pane.
- Click  to open the **User** showing the user list.
- Select a user to delete in the object pane.
- Click the  **Delete User** from the object pane toolbar or right-click the user and select  **Delete User** from the popup menu.
- Confirm deleting in the dialog window.

## Privilege Manager

To edit privilege according to the database objects by using Privilege Manager

- Select the connection you wish to set privileges in the navigation pane.
- Click  to open the **User** showing the user list.
- Click the  **Privilege Manager** to open the **Privilege Manager** window and set privileges.

## Privileges Provided by MySQL

The primary function of the MySQL privilege system is to authenticate a user who connects from a given host and to associate that user with privileges on a database such as *SELECT*, *INSERT*, *UPDATE*, and *DELETE*.

Information about user privileges is stored in the **user**, **db**, **host**, **tables\_priv**, **columns\_priv**, and **procs\_priv** tables in the **mysql** database (that is, in the database named mysql). The MySQL server reads the contents of these tables when it starts.

MySQL access control involves two stages when you run a client program that connects to the server:

- Stage 1: The server checks whether it should allow you to connect.
- Stage 2: Assuming that you can connect, the server checks each statement you issue to determine whether you have sufficient privileges to perform it. For examples:  
Create table privilege, Drop table privilege or Alter table privilege.

The server uses the **user**, **db**, and **host** tables in the **mysql** database at both stages of access control.

## MySQL User Designer

The **User Designer** window allows you to set different properties and privileges for a MySQL user.

- [Editing User General](#)
- [Setting Advanced User Properties](#)
- [Setting Server Privileges](#)
- [Setting Privileges](#)
- SQL Preview



## Editing MySQL User General

The **General** tab allows you to set user properties which are **User name**, **Host** and **Password**.

## Setting Advanced MySQL User Properties

### Max queries per hour, Max updates per hour and Max connections per hour

These options limit the number of queries, updates, and logins a user can perform during any given one-hour period. If they are set as 0 (the default), this means that there is no limitation for that user.

### Max user connections

This option limits the maximum number of simultaneous connections that the account can make. If it is set as 0 (the default), the *max\_user\_connections* system variable determines the number of simultaneous connections for the account.

### ☒ Use OLD\_PASSWORD encryption

The password hashing mechanism was updated in MySQL 4.1 to provide better security and to reduce the risk of passwords being intercepted. However, this new mechanism is understood only by MySQL 4.1 (and newer) servers and clients, which can result in some compatibility problems. A 4.1 or newer client can connect to a pre-4.1 server, because the client understands both the old and new password hashing mechanisms. However, a pre-4.1 client that attempts to connect to a 4.1 or newer server may run into difficulties.

Enable this option if you wish to maintain backward compatibility with pre-4.1 clients under circumstances where the server would otherwise generate long password hashes. The option does not affect authentication (4.1 and later clients can still use accounts that have long password hashes), but it does prevent creation of a long password hash in the *user* table as the result of a password-changing operation.

## SSL

MySQL can check X509 certificate attributes in addition to the usual authentication that is based on the username and password. To specify SSL-related options for a MySQL account, use the *REQUIRE* clause of the *GRANT* statement.

### ANY

This option tells the server to allow only SSL-encrypted connections for the account.

Example:

```
GRANT ALL PRIVILEGES ON test.* TO 'root'@'localhost'  
IDENTIFIED BY 'goodsecret' REQUIRE SSL;
```

## X509

This means that the client must have a valid certificate but that the exact certificate, issuer, and subject do not matter. The only requirement is that it should be possible to verify its signature with one of the CA certificates.

Example:

```
GRANT ALL PRIVILEGES ON test.* TO 'root'@'localhost'
IDENTIFIED BY 'goodsecret' REQUIRE SSL;
```

## SPECIFIED

Example:

```
GRANT ALL PRIVILEGES ON test.* TO 'root'@'localhost'
IDENTIFIED BY 'goodsecret'
REQUIRE SUBJECT '/C=EE/ST=Some-State/L=Tallinn/
O=MySQL demo client certificate/
CN=Tonu Samuel/Email=tonu@example.com'
AND ISSUER '/C=FI/ST=Some-State/L=Helsinki/
O=MySQL Finland AB/CN=Tonu Samuel/Email=tonu@example.com'
AND CIPHER 'EDH-RSA-DES-CBC3-SHA';
```

### Issuer

This places the restriction on connection attempts that the client must present a valid X509 certificate issued by CA *issuer*. If the client presents a certificate that is valid but has a different issuer, the server rejects the connection. Use of X509 certificates always implies encryption, so the SSL option is unnecessary in this case.

### Subject

This places the restriction on connection attempts that the client must present a valid X509 certificate containing the subject *subject*. If the client presents a certificate that is valid but has a different subject, the server rejects the connection.

### Cipher


This is needed to ensure that ciphers and key lengths of sufficient strength are used. SSL itself can be weak if old algorithms using short encryption keys are used. Using this option, you can ask that a specific cipher method is used to allow a connection.

## Setting MySQL User Server Privileges

In the grid, check **Granted** option against the server privilege listed in **Privilege** to assign this user to have that privilege. Multiple privileges can be granted.


To grant (select) or revoke (unselect) all privileges, right-click the grid and select **Grant All** or **Revoke All** option.

## Setting MySQL User Privileges

To edit the specific object privileges of the user, click  **Add Privilege** to open the window and follow the steps below:

- (1) Expand the node in the tree view until reaching to the target object.
- (2) Check the object to show the grid on the right panel.
- (3) In the grid, check **State** option against the privilege listed in **Privilege** to assign this user to have that privilege. Multiple privileges can be granted.

To grant (select) or revoke (unselect) all privileges, right-click the grid and select **Grant All** or **Revoke All** option.

**Note:** Click  **Save** to apply any changes you have made.


## Oracle Security Management

Oracle manages database access permissions using users and roles. Users own schema objects (for example, tables, views) and can assign privileges on those objects to other users to control who has access to which objects.

Navicat provides **User** to add, duplicate, edit, delete users/roles, grant/revoke server privileges and privileges on the selected schema objects. The object pane displays all the users/roles that exist in the server.




In addition to the user accounts that you create, the database includes a number of user accounts that are automatically created upon installation. Administrative accounts: **SYS**, **SYSTEM**, **SYSMAN**, and **DBSNMP**. Administrative accounts are highly privileged accounts to perform administrative tasks such as starting and stopping the database, managing database memory and storage, creating and managing database users, and so on. Your database may also include sample schemas (**SCOTT**, **HR**, **OE**, **OC**, **PM**, **IX** and **SH**), which are a set of interlinked schemas that enable Oracle documentation and Oracle instructional materials to illustrate common database tasks.

### Manage User

Just simply click  -> **User** to open an object pane for **User**. A right-click displays the popup menu or use the object pane toolbar, allowing you to add, edit and delete users.


### Add User

To add a new user

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **User** to open the **User** showing the user list.
- Click the  **New User** from the object pane toolbar or right-click and select  **New User** from the popup menu.
- Edit user properties and privileges on the appropriate tabs of the User Designer.




## Duplicate User

To create a new user with modification as one of the existing users

- Select the connection you wish to set privileges in the navigation pane.
- Click -> **User** to open the **User** showing the user list.
- Select a user to edit in the object pane.
- Right-click the user and select **Duplicate User** from the popup menu.
- Edit user properties and privileges on the appropriate tabs of the User Designer.




## Edit User

To edit an existing user


- Select the connection you wish to set privileges in the navigation pane.
- Click -> **User** to open the **User** showing the user list.
- Select a user to edit in the object pane.
- Click the  **Edit User** from the object pane toolbar or right-click the user and select  **Edit User** from the popup menu.
- Edit user properties and privileges on the appropriate tabs of the User Designer.

## Delete User

To delete a user




- Select the connection you wish to set privileges in the navigation pane.
- Click -> **User** to open the **User** showing the user list.
- Select a user to delete in the object pane.
- Click the  **Delete User** from the object pane toolbar or right-click the user and select  **Delete User** from the popup menu.
- Confirm deleting in the dialog window.

## Manage Role

Just simply click  -> **Role** to open an object pane for **Role**. A right-click displays the popup menu or use the object pane toolbar, allowing you to add, edit and delete roles.


## Add Role

To add a new role

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Role** to open the **Role** showing the role list.
- Click the  **New Role** from the object pane toolbar or right-click and select  **New Role** from the popup menu.
- Edit role properties and privileges on the appropriate tabs of the Role Designer.




## Duplicate Role

To create a new role with modification as one of the existing roles

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Role** to open the **Role** showing the role list.
- Select a role to edit in the object pane.
- Right-click the role and select **Duplicate Role** from the popup menu.
- Edit role properties and privileges on the appropriate tabs of the Role Designer.

## Edit Role




To edit an existing role

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Role** to open the **Role** showing the role list.
- Select a role to edit in the object pane.
- Click the  **Edit Role** from the object pane toolbar or right-click the role and select  **Edit Role** from the popup menu.
- Edit role properties and privileges on the appropriate tabs of the Role Designer.





## Delete Role

To delete a role

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Role** to open the **Role** showing the role list.
- Select a role to delete in the object pane.
- Click the  **Delete Role** from the object pane toolbar or right-click the role and select  **Delete Role** from the popup menu.
- Confirm deleting in the dialog window.

## Privilege Manager

To edit privilege according to the database objects by using Privilege Manager

- Select the connection you wish to set privileges in the navigation pane.
- Click  to open either one showing the user/role list.
- Click the  **Privilege Manager** to open the **Privilege Manager** window and set privileges.

## Privileges Provided by Oracle

In Oracle, a set of access privileges and restrictions exist for each applicable database object.

When you create a database object, you become its owner. By default, only the owner of an object can do anything with the object. In order to allow other users to use it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

Ordinarily, only the object's owner (or a superuser) can grant or revoke privileges on an object. However, it is possible to grant a privilege **Admin Option/Grant Option**, which gives the recipient the right to grant it in turn to others. If the grant option is subsequently revoked then all who received the privilege from that recipient (directly or through a chain of grants) will lose the privilege.

**Note:** The special name **PUBLIC** is accessible to every database user, all privileges and roles granted to **PUBLIC** are accessible to every database user.

## Oracle User Designer

The **User Designer** window allows you to set different properties and privileges for a Oracle user.

- [Editing User General](#)
- [Setting User Membership](#)
- [Setting User Quotas](#)
- [Setting Server Privileges](#)
- [Setting Privileges](#)
- SQL Preview

## Editing Oracle User General

The **General** tab allows you to set user properties which are:

### User name

Set name of the user.

### Authentication

Choose to use either Password, External or Global as authentication method.

#### Password

A local user must specify password to log on to the database.

##### Password

Set user's password.

##### Confirm Password

Re-type the user's password here.

##### ☒ Expire Password

Expire the user's password. This setting forces the user or the DBA to change the password before the user can log in to the database.

#### External

An external user must be authenticated by an external service, such as an operating system or a third-party service.

#### Global

A global user must be authorized by the enterprise directory service (Oracle Internet Directory).

#### X.500 Name

Enter the X.509 name at the enterprise directory service that identifies this user.

### Default Table Space

Choose the default tablespace for objects that the user creates.

### Temporary Table Space

Choose the tablespace or tablespace group for the user's temporary segments.

## Profile

Choose the profile that assign to the user.

### ☒ Lock Account

Lock the user's account and disable access.

## Setting Oracle User Membership

In the grid, check **Granted**, **Admin Option** or **As Default** option against the role listed in **Role Name** to assign this user to be a member of selected role. Multiple roles can be granted.

## Setting Oracle User Quotas

In the grid, specify the maximum amount of space the user can allocate in the tablespaces. Enter the **Quota** and choose the **Unit** of the **Tablespace**. **Unlimited** lets the user allocate space in the tablespace without bound. Multiple tablespaces can be set.


## Setting Oracle User Server Privileges

In the grid, check **Granted** or **Admin Option** option against the server privilege listed in **Privilege** to assign this user to have that privilege. Multiple privileges can be granted.

To grant (select) or revoke (unselect) all privileges, right-click the grid and select **Grant All**, **Grant All with Grant Option** or **Revoke All** option.




## Setting Oracle User Privileges

To edit the specific object privileges of the user, click  **Add Privilege** to open the window and follow the steps below:

- (1) Expand the node in the tree view until reaching to the target object.
- (2) Check the object to show the grid on the right panel.
- (3) In the grid, check **Granted** or **Grant Option** option against the privilege listed in **Privilege** to assign this user to have that privilege. Multiple privileges can be granted.

To grant (select) or revoke (unselect) all privileges, right-click the grid and select **Grant All**, **Grant All with Grant Option** or **Revoke All** option.

**Note:** Click  **Save** to apply any changes you have made.

## Oracle Role Designer

The **Role Designer** window allows you to set different properties and privileges for a Oracle role.

- [Editing Role General](#)
- [Setting Role Membership](#)
- [Setting Role Members](#)
- [Setting Server Privileges](#)
- [Setting Privileges](#)
- SQL Preview

## Editing Oracle Role General

The **General** tab allows you to set role properties which are:

### **Role name**

Set name of the role.

### **Authentication**

Choose to use either Password, External or Global Authentication method.

#### **Password**

User must specify the password to the database when enabling the role.

##### **Password**

Set role's password.

##### **Confirm Password**

Re-type the role's password here.

#### **External**

An external user must be authorized by an external service, such as an operating system or third-party service, before enabling the role.

#### **Global**

A global user must be authorized to use the role by the enterprise directory service before the role is enabled at login.

#### **Not Identified**

The role is authorized by the database and that no password is required to enable the role.

## Setting Oracle Role Membership

In the grid, check **Granted** or **Admin Option** option against the role listed in **Role Name** to assign this role to be a member of selected role. Multiple roles can be granted.

## Setting Oracle Role Members


In the grid, check **Granted** or **Admin Option** option against user listed in **Member** to assign the selected user to be a member of this role. Multiple users can be granted.

## Setting Oracle Role Server Privileges

In the grid, check **Granted** or **Admin Option** option against the server privilege listed in **Privilege** to assign this role to have that privilege. Multiple privileges can be granted.


To grant (select) or revoke (unselect) all privileges, right-click the grid and select **Grant All**, **Grant All with Grant Option** or **Revoke All** option.

## Setting Oracle Role Privileges

To edit the specific object privileges of the role, click  **Add Privilege** to open the window and follow the steps below:

- (1) Expand the node in the tree view until reaching to the target object.
- (2) Check the object to show the grid on the right panel.
- (3) In the grid, check **Grant** option against the privilege listed in **Privilege** to assign this role to have that privilege. Multiple privileges can be granted.

To grant (select) or revoke (unselect) all privileges, right-click the grid and select **Grant All** or **Revoke All** option.

**Note:** Click  **Save** to apply any changes you have made.

## PostgreSQL Security Management

PostgreSQL manages database access permissions using users and groups. Users own database objects (for example, tables) and can assign privileges on those objects to other users to control who has access to which objects.

**Note:** Starting from PostgreSQL version 8.1, users and groups were no longer distinct kinds of entities, now there are only roles. Any role can act as a user, a group, or both. The concept of roles subsumes the concepts of users and groups.


Navicat provides **User** to add, duplicate, edit, delete users/groups/roles, grant/revoke server privileges and privileges on the selected database objects. The object pane displays all the users/groups/roles that exist in the server.

Only a superuser (a user who is allowed all rights) can add/delete users. PostgreSQL installs a single superuser by default named **postgres**. All other users must be added by this user, or by another subsequently added superuser.

The **User** for PostgreSQL Server 7.3 to 8.0 and PostgreSQL Server 8.1 to 9.1 are different.




### PostgreSQL Server 7.3 to 8.0

#### Manage User

Just simply click  -> **User** to open an object pane for **User**. A right-click displays the popup menu or use the object pane toolbar, allowing you to add, edit and delete users.

#### Add User


To add a new user

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **User** to open the **User** showing the user list.
- Click the  **New User** from the object pane toolbar or right-click and select  **New User** from the popup menu.
- Edit user properties and privileges on the appropriate tabs of the User Designer.






## Duplicate User

To create a new user with modification as one of the existing users

- Select the connection you wish to set privileges in the navigation pane.
- Click -> **User** to open the **User** showing the user list.
- Select a user to edit in the object pane.
- Right-click the user and select **Duplicate User** from the popup menu.
- Edit user properties and privileges on the appropriate tabs of the User Designer.




## Edit User

To edit an existing user


- Select the connection you wish to set privileges in the navigation pane.
- Click -> **User** to open the **User** showing the user list.
- Select a user to edit in the object pane.
- Click the  **Edit User** from the object pane toolbar or right-click the user and select  **Edit User** from the popup menu.
- Edit user properties and privileges on the appropriate tabs of the User Designer.

## Delete User

To delete a user




- Select the connection you wish to set privileges in the navigation pane.
- Click -> **User** to open the **User** showing the user list.
- Select a user to delete in the object pane.
- Click the  **Delete User** from the object pane toolbar or right-click the user and select  **Delete User** from the popup menu.
- Confirm deleting in the dialog window.

## Manage Group

Just simply click  -> **Group** to open an object pane for **Group**. A right-click displays the popup menu or use the object pane toolbar, allowing you to add, edit and delete groups.


## Add Group

To add a new group

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Group** to open the **Group** showing the group list.
- Click the  **New Group** from the object pane toolbar or right-click and select  **New Group** from the popup menu.
- Edit group properties and privileges on the appropriate tabs of the Group Designer.




## Duplicate Group

To create a new group with modification as one of the existing groups

- Select the connection you wish to set privileges in the navigation pane.
- Click  to open the **Group** showing the group list.
- Select a group to edit in the object pane.
- Right-click the group and select **Duplicate Group** from the popup menu.
- Edit group properties and privileges on the appropriate tabs of the Group Designer.




## Edit Group

To edit an existing group


- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Group** to open the **Group** showing the group list.
- Select a group to edit in the object pane.
- Click the  **Edit Group** from the object pane toolbar or right-click the group and select  **Edit Group** from the popup menu.
- Edit group properties and privileges on the appropriate tabs of the Group Designer.

## Delete Group

To delete a group




- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Group** to open the **Group** showing the group list.
- Select a group to delete in the object pane.
- Click the  **Delete Group** from the object pane toolbar or right-click the group and select  **Delete Group** from the popup menu.
- Confirm deleting in the dialog window.

## PostgreSQL Server 8.1 to 9.1

Just simply click  -> **Role** to open an object pane for **Role**. A right-click displays the popup menu or use the object pane toolbar, allowing you to add, edit and delete roles.


## Add Role

To add a new role

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Role** to open the **Role** showing the role list.
- Click the  **New Role** from the object pane toolbar or right-click and select  **New Role** from the popup menu.
- Edit role properties and privileges on the appropriate tabs of the Role Designer.




## Duplicate Role

To create a new role with modification as one of the existing roles

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Role** to open the **Role** showing the role list.
- Select a role to edit in the object pane.
- Right-click the role and select **Duplicate Role** from the popup menu.
- Edit role properties and privileges on the appropriate tabs of the Role Designer.




## Edit Role

To edit an existing role

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Role** to open the **Role** showing the role list.
- Select a role to edit in the object pane.
- Click the  **Edit Role** from the object pane toolbar or right-click the role and select  **Edit Role** from the popup menu.
- Edit role properties and privileges on the appropriate tabs of the Role Designer.



## Delete Role

To delete a role

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Role** to open the **Role** showing the role list.
- Select a role to delete in the object pane.
- Click the  **Delete Role** from the object pane toolbar or right-click the role and select  **Delete Role** from the popup menu.
- Confirm deleting in the dialog window.

## Privilege Manager

To edit privilege according to the database objects by using Privilege Manager

- Select the connection you wish to set privileges in the navigation pane.
- Click  to open either one showing the user/group/role list.
- Click the  **Privilege Manager** to open the **Privilege Manager** window and set privileges.

## Privileges Provided by PostgreSQL

In PostgreSQL, a set of access privileges and restrictions exist for each applicable database object.

When you create a database object, you become its owner. By default, only the owner of an object can do anything with the object. In order to allow other users to use it, privileges must be granted. (However, users that have the superuser attribute can always access any object.)

Different privileges: *SELECT*, *INSERT*, *UPDATE*, *DELETE*, *REFERENCES*, *TRIGGER*, *CREATE*, *CONNECT*, *TEMPORARY*, *EXECUTE*, and *USAGE*. The privileges applicable to a particular object vary depending on the object's type (table, function, etc).

Ordinarily, only the object's owner (or a superuser) can grant or revoke privileges on an object. However, it is possible to grant a privilege **With Grant Option**, which gives the recipient the right to grant it in turn to others. If the grant option is subsequently revoked then all who received the privilege from that recipient (directly or through a chain of grants) will lose the privilege.

**Note:** The special name **public** can be used to grant a privilege to every role (user/group) on the system.

## **Manage Users for PostgreSQL Server 7.3 to 8.0**

PostgreSQL version 7.3 to 8.0 manages database access permissions using users and groups.

- [User Designer](#)
- [Group Designer](#)

## PostgreSQL User Designer

The **User Designer** window allows you to set different properties and privileges for a PostgreSQL user.

- [Editing User General](#)
- [Setting User Membership](#)
- [Setting Privileges](#)
- SQL Preview

## Editing PostgreSQL User General

The **General** tab allows you to set user properties which are:

### User Name

Set name of the user.

### User ID

Specify an ID for the user. This is normally not necessary, but may be useful if you need to recreate the owner of an orphaned object. If this is not specified, the highest assigned user ID plus one (with a minimum of 100) will be used as default.

### Password

Set user's password.

**Note:** If you do not plan to use password authentication you can omit this option, but then the user will not be able to connect if you decide to switch to password authentication.

### Confirm Password

Re-type the password here.

### Password Encryption

This option control whether the password is stored **ENCRYPTED** or **UNENCRYPTED** in the system catalogs. (If neither is specified, the default behavior is determined by the configuration parameter *password\_encryption*.)

### Expiry Date

Set a date and time after which the user's password is no longer valid. If this clause is omitted the password will be valid for all time.

### ☒ Superuser

Check this option to define the user as a superuser.

### ☒ Can create database


Check this option to define the user to be allowed to create databases.



## Setting PostgreSQL User Membership


In the grid, check **Granted** option against the group listed in **Group Name** to assign this user to be a member of selected group. Multiple groups can be granted.

## Setting PostgreSQL User Privileges

To edit the specific object privileges of the user, click  **Add Privilege** to open the window and follow the steps below:

- (1) Expand the node in the tree view until reaching to the target object.
- (2) Check the object to show the grid on the right panel.
- (3) In the grid, check **Granted** or **Grant Option** option against the privilege listed in **Privilege** to assign this user to have that privilege. Multiple privileges can be granted.

To grant (select) or revoke (unselect) all privileges, right-click the grid and select **Grant All**, **Grant All with Grant Option** or **Revoke All** option.

**Note:** Click  **Save** to apply any changes you have made.

## PostgreSQL Group Designer

The **Group Designer** window allows you to set different properties and privileges for a PostgreSQL group.

- [Editing Group General](#)
- [Setting Group Members](#)
- [Setting Privileges](#)
- SQL Preview

## Editing PostgreSQL Group General

The **General** tab allows you to set group properties which are:

### **Group name**

Set name of the group.


### **Group ID**

Specify an ID for the group. This is normally not necessary, but may be useful if you need to recreate a group referenced in the permissions of some object. If this is not specified, the highest assigned group ID plus one (with a minimum of 100) will be used as default.

## Setting PostgreSQL Group Members


In the grid, check **Granted** option against the user listed in **Member** to assign selected user to be a member of this group. Multiple users can be granted.

## Setting PostgreSQL Group Privileges

To edit the specific object privileges of the group, click  **Add Privilege** to open the window and follow the steps below:

- (1) Expand the node in the tree view until reaching to the target object.
- (2) Check the object to show the grid on the right panel.
- (3) In the grid, check **Grant** option against the privilege listed in **Privilege** to assign this group to have that privilege. Multiple privileges can be granted.

To grant (select) or revoke (unselect) all privileges, right-click the grid and select **Grant All** or **Revoke All** option.

**Note:** Click  **Save** to apply any changes you have made.

## **Manage Users for PostgreSQL Server 8.1 to 9.1**

Starting from PostgreSQL version 8.1, users and groups were no longer distinct kinds of entities, now there are only roles. Any role can act as a user, a group, or both. The concept of roles subsumes the concepts of users and groups.

- [Role Designer](#)

## PostgreSQL Role Designer

The **Role Designer** window allows you to set different properties and privileges for a PostgreSQL role.

- [Editing Role General](#)
- [Setting Role Membership](#)
- [Setting Role Members](#)
- [Setting Privileges](#)
- SQL Preview



## Editing PostgreSQL Role General

The **General** tab allows you to set role properties which are:

### Role Name

Set name of the role.

### Role ID

Specify an ID for the role. This is normally not necessary, but may be useful if you need to recreate the owner of an orphaned object. If this is not specified, the highest assigned role ID plus one (with a minimum of 100) will be used as default.

**Note:** In PostgreSQL versions 8.1 or above, the specified ID will be ignored, but is accepted for backwards compatibility.

### ☒ Can login

Check this option to create a role that allow to login. A role having this option can be thought of as a user. Roles without this attribute are useful for managing database privileges, but are not users in the usual sense of the word.

### Password

Set role's password.

**Note:** If you do not plan to use password authentication you can omit this option, but then the role will not be able to connect if you decide to switch to password authentication.

### Confirm Password

Re-type the password here.

### Password Encryption

This option control whether the password is stored **ENCRYPTED** or **UNENCRYPTED** in the system catalogs. (If neither is specified, the default behavior is determined by the configuration parameter *password\_encryption*.)

### Connection Limit

If role can log in, this specifies how many concurrent connections the role can make. -1 (the default) means no limit.

## **Expiry Date**

Set a date and time after which the role's password is no longer valid. If this clause is omitted the password will be valid for all time.

## ☒ **Superuser**

Check this option to determine the new role is a superuser, who can override all access restrictions within the database.

## ☒ **Can create database**

Check this option to define a role's ability to create databases.

## ☒ **Can create role**

Check this option to allow creating roles.

## ☒ **Inherit privileges**

Check this option to determine whether a role inherits the privileges of roles it is a member of.

## ☒ **Can update system catalog**

Check this option to allow a role's ability to update system catalog.


## Setting PostgreSQL Role Membership

In the grid, check **Granted** or **Admin Option** option against the role listed in **Role Name** to assign this role to be a member of selected role. Multiple roles can be granted.

## Setting PostgreSQL Role Members


In the grid, check **Granted** or **Admin Option** option against the role listed in **Member** to assign the selected role to be a member of this role. Multiple roles can be granted.

## Setting PostgreSQL Role Privileges

To edit the specific object privileges of the role, click  **Add Privilege** to open the window and follow the steps below:

- (1) Expand the node in the tree view until reaching to the target object.
- (2) Check the object to show the grid on the right panel.
- (3) In the grid, check **Granted** or **Grant Option** option against the privilege listed in **Privilege** to assign this role to have that privilege. Multiple privileges can be granted.

To grant (select) or revoke (unselect) all privileges, right-click the grid and select **Grant All**, **Grant All with Grant Option** or **Revoke All** option.

**Note:** Click  **Save** to apply any changes you have made.


## SQL Server Security Management

Navicat provides **User** to add, duplicate, edit, delete users/roles, grant/revoke server permissions and permissions on the selected database objects. The object pane displays all the users/roles that exist in the server.

The SQL Server **sa** log in is a server-level principal. By default, it is created when an instance is installed. In SQL Server 2005 or above, the default database of **sa** is **master**. This is a change of behavior from earlier versions of SQL Server.




By default, the database includes a **guest** user when a database is created. Permissions granted to the **guest** user are inherited by users who do not have a user account in the database. The **guest** user cannot be dropped, but it can be disabled by revoking its CONNECT permission. The CONNECT permission can be revoked by executing REVOKE CONNECT FROM GUEST within any database other than **master** or **tempdb**.

### Manage Login

Just simply click  -> **Login** to open an object pane for **Login**. A right-click displays the popup menu or use the object pane toolbar, allowing you to add, edit and delete login.


### Add Login

To add a new login

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Login** to open the **Login** showing the login list.
- Click the  **New Login** from the object pane toolbar or right-click and select  **New Login** from the popup menu.
- Edit login properties and permissions on the appropriate tabs of the Login Designer.




### Duplicate Login

To create a new login with modification as one of the existing logins

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Login** to open the **Login** showing the login list.
- Select a login to edit in the object pane.
- Right-click the login and select **Duplicate Login** from the popup menu.
- Edit login properties and privileges on the appropriate tabs of the Login Designer.




## Edit Login

To edit an existing login


- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Login** to open the **Login** showing the login list.
- Select a login to edit in the object pane.
- Click the  **Edit Login** from the object pane toolbar or right-click the login and select  **Edit Login** from the popup menu.
- Edit login properties and permissions on the appropriate tabs of the Login Designer.

## Delete Login




To delete a login

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Login** to open the **Login** showing the login list.
- Select a login to delete in the object pane.
- Click the  **Delete Login** from the object pane toolbar or right-click the login and select  **Delete Login** from the popup menu.
- Confirm deleting in the dialog window.


## Manage Server Role

Just simply click  -> **Server Role** to open an object pane for **Server Role**. A right-click displays the popup menu or use the object pane toolbar, allowing you to edit server role.

To edit an existing server role




- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Server Role** to open the **Server Role** showing the server role list.
- Select a server role to edit in the object pane.
- Click the  **Edit Server Role** from the object pane toolbar or right-click the server role and select  **Edit Server Role** from the popup menu.
- Edit server role properties and permissions on the appropriate tabs of the Server Role Designer.

## Manage Database User

Just simply click  -> **Database User** to open an object pane for **Database User**. A right-click displays the popup menu or use the object pane toolbar, allowing you to add, edit and delete database users.


### Add Database User

To add a new database user

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Database User** to open the **Database User** showing the database user list.
- Click the  **New Database User** from the object pane toolbar or right-click and select  **New Database User** from the popup menu.
- Edit database user properties and permissions on the appropriate tabs of the Database User Designer.

### Duplicate Database User




To create a new database user with modification as one of the existing database users

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Database User** to open the **Database User** showing the database user list.
- Select a database user to edit in the object pane.
- Right-click the database user and select **Duplicate Database User** from the popup menu.
- Edit database user properties and privileges on the appropriate tabs of the Database User Designer.






## Edit Database User

To edit an existing database user


- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Database User** to open the **Database User** showing the database user list.
- Select a database user to edit in the object pane.
- Click the  **Edit Database User** from the object pane toolbar or right-click the database user and select  **Edit Database User** from the popup menu.
- Edit database user properties and permissions on the appropriate tabs of the Database User Designer.

## Delete Database User

To delete a database user




- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Database User** to open the **Database User** showing the database user list.
- Select a database user to delete in the object pane.
- Click the  **Delete Database User** from the object pane toolbar or right-click the database user and select  **Delete Database User** from the popup menu.
- Confirm deleting in the dialog window.

## Manage Database Role

Just simply click  -> **Database Role** to open an object pane for **Database Role**. A right-click displays the popup menu or use the object pane toolbar, allowing you to add, edit and delete database roles.


## Add Database Role

To add a new database role

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Database Role** to open the **Database Role** showing the database role list.
- Click the  **New Database Role** from the object pane toolbar or right-click and select  **New Database Role** from the popup menu.
- Edit database role properties and permissions on the appropriate tabs of the Database Role Designer.




## Duplicate Database Role

To create a new database role with modification as one of the existing database roles

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Database Role** to open the **Database Role** showing the database role list.
- Select a database role to edit in the object pane.
- Right-click the database role and select **Duplicate Database Role** from the popup menu.
- Edit database role properties and privileges on the appropriate tabs of the Database Role Designer.




## Edit Database Role

To edit an existing database role


- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Database Role** to open the **Database Role** showing the database role list.
- Select a database role to edit in the object pane.
- Click the  **Edit Database Role** from the object pane toolbar or right-click the database role and select  **Edit Database Role** from the popup menu.
- Edit database role properties and permissions on the appropriate tabs of the Database Role Designer.

## Delete Database Role

To delete a Database Role




- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Database Role** to open the **Database Role** showing the database role list.
- Select a database role to delete in the object pane.
- Click the  **Delete Database Role** from the object pane toolbar or right-click the database role and select  **Delete Database Role** from the popup menu.
- Confirm deleting in the dialog window.

## Manage Application Role

Just simply click  -> **Application Role** to open an object pane for **Application Role**. A right-click displays the popup menu or use the object pane toolbar, allowing you to add, edit and delete application roles.


## Add Application Role

To add a new application role

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Application Role** to open the **Application Role** showing the application role list.
- Click the  **New Application Role** from the object pane toolbar or right-click and select  **New Application Role** from the popup menu.
- Edit application role properties and permissions on the appropriate tabs of the Application Role Designer.




## Duplicate Application Role

To create a new application role with modification as one of the existing application roles

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Application Role** to open the **Application Role** showing the application role list.
- Select an application role to edit in the object pane.
- Right-click the application role and select **Duplicate Application Role** from the popup menu.
- Edit application role properties and privileges on the appropriate tabs of the Application Role Designer.




## Edit Application Role

To edit an existing application role

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Application Role** to open the **Application Role** showing the application role list.
- Select an application role to edit in the object pane.
- Click the  **Edit Application Role** from the object pane toolbar or right-click the application role and select  **Edit Application Role** from the popup menu.
- Edit application role properties and permissions on the appropriate tabs of the Application Role Designer.



## Delete Application Role

To delete an application role

- Select the connection you wish to set privileges in the navigation pane.
- Click  -> **Application Role** to open the **Application Role** showing the application role list.
- Select an application role to delete in the object pane.
- Click the  **Delete Application Role** from the object pane toolbar or right-click the application role and select  **Delete Application Role** from the popup menu.
- Confirm deleting in the dialog window.

## Privilege Manager

To edit privilege according to the database objects by using Privilege Manager

- Select the connection you wish to set privileges in the navigation pane.
- Click  to open either one showing the login/server role/database user/database role/application role list.
- Click the  **Privilege Manager** to open the **Privilege Manager** window and set privileges.

## Privileges Provided by SQL Server

In SQL Server, the concept for permissions is using principals and securables. Principals are the individuals, groups, and processes granted access to SQL Server. Securables are the server, database, and objects the database contains. Principals can be arranged in a hierarchy. To easily manage the permissions in your databases, SQL Server provides several roles which are security principals that group other principals. Database-level roles are database-wide in their permissions scope.

### Windows-level principals

- Windows Domain Login
- Windows Local Login

### SQL Server-level principal

- SQL Server Login

### Database-level principals

- Database User
- Database Role
- Application Role

## Login

SQL Server uses two ways to validate connections to SQL Server databases: Windows Authentication and SQL Server Authentication. SQL Server Authentication uses login records to validate the connection. A Login object exposes a SQL Server login record.

## Server Role

Server-level roles are also named fixed server roles because you cannot create new server-level roles and the permissions of fixed server roles cannot be changed. You can add SQL Server logins, Windows accounts, and Windows groups into server-level roles. Each member of a fixed server role can add other logins to that same role.

## Database User

To gain access to a database, a login must be identified as a database user. The database user is usually known by the same name as the login, but you can create a database user (for a login) with a different name.

## **Database Role**

Fixed database roles are defined at the database level and exist in each database. You can add any database account and other SQL Server roles into database-level roles. Each member of a fixed database role can add other logins to that same role.

## **Application Role**

An application role is a database principal that enables an application to run with its own, user-like permissions. You can use application roles to enable access to specific data to only those users who connect through a particular application. Unlike database roles, application roles contain no members and are inactive by default.

## SQL Server Login Designer

The **Login Designer** window allows you to set different properties and privileges for a SQL Server login.

- [Editing Login General](#)
- [Setting Roles](#)
- [Setting User Mapping](#)
- [Setting Server Permissions](#)
- [Setting Endpoint Permissions](#)
- [Setting Login Permissions](#)
- SQL Preview



## Editing SQL Server Login General

### Options for SQL Server

#### Login Name

Set name of the login.

#### Authentication Type

Select the authentication type.

#### SQL Server Authentication

Selects to use the SQL Server login for authentication.

#### Password

A login must specify password to log on to the database.

#### Confirm Password

Re-type the login's password here.

#### ☒ Specify Old Password

Check this option to enter the old password used by this account.

#### ☒ Enforce Password Policy

You can check this option to force password to follow password policy of SQL Server.

**Note:** Support from SQL Server 2005 or later.

#### ☒ Enforce Password Expiration

You can check this option to force password to have expiry date.

**Note:** Support from SQL Server 2005 or later.

#### ☒ User Must Change Password at Next Login

You can check this option to force user to change password everytime when login.

**Note:** Support from SQL Server 2005 or later.

#### Windows Authentication

Selects to use the Windows login for authentication.

## **Mapped to Certificate**

You can select to use certificate for authentication.

SQL Server contains features that enable you to create and manage certificates and keys for use with the server and database. You can use externally generated certificates or SQL Server can generate certificates.

**Note:** Support from SQL Server 2005 or later.

### **Certificate Name**

Select the certificate name.

## **Mapped to Asymmetric Key**

You can select to use asymmetric key for authentication.

**Note:** Support from SQL Server 2005 or later.

### **Asymmetric Key Name**

Selects the asymmetric key name.

**Note:** Certificates and asymmetric keys are both ways to use asymmetric encryption. There is no difference between the two mechanisms for the cryptographic algorithm, and no difference in strength given the same key length.

## **Default Database**

Selects the default database when login.

## **Default Language**

Selects the default display language when login.

### ☒ **Enabled**

Checks to enable the login.

## **Credential**

You can add credential on specific role for this login. A credential is a record that contains the authentication information (credentials) required to connect to a resource outside SQL Server. This information is used internally by SQL Server.

**Note:** Support from SQL Server 2005 or later.

## Options for SQL Azure

### **Login Name**

Set name of the login.

### **Password**

A login must specify password to log on to the database.

### **Confirm Password**

Re-type the login's password here.

### ☒ **Enabled**

Checks to enable the login.

## Setting SQL Server Login Roles

In the grid, check the server role to assign this server login to be a member of selected server role. Multiple roles can be granted.

**Note:** Every SQL Server login belongs to the **public** server role. When a server principal has not been granted or denied specific permissions on a securable object, the user inherits the permissions granted to **public** on that object. Only assign **public** permissions on any object when you want the object to be available to all users.

**Note:** SQL Azure does not support.

## Setting SQL Server Login User Mapping

In the Grid, check the **Database** and enter the **User** and **Default Schema** to create user for login the database and specify the first schema will be searched by the server.

**Note:** SQL Azure does not support.

## Setting SQL Server Login Server Permissions

You can check **Grant**, **With Grant Option** or **Deny** against the server permissions listed in **Permission** to assign this login to have that permission. Multiple permissions can be granted.

To grant (select) or revoke (unselect) all privileges, right-click the grid and select **Grant All**, **Grant All with Grant Option**, **Deny All** or **Revoke All** option.

**Note:** Support from SQL Server 2005 or later.

## Setting SQL Server Login Endpoint Permissions

You can check **Alter**, **Connect**, **Control**, **Take Ownership** or **View Definition** against the endpoint listed in **Endpoint** to assign this login to have that endpoint permission. Multiple permissions can be granted. You can click on the checkbox to have more choices on the permission setting.

**Note:** Support from SQL Server 2005 or later.

## Setting SQL Server Login Login Permissions

You can check **Alter**, **Control**, **Impersonate** or **View Definition** against the server login listed in **Login** to assign this server login to have that login permission. Multiple permissions can be granted. You can click on the checkbox to have more choices on the permission setting.

**Note:** Support from SQL Server 2005 or later.



## SQL Server Server Role Designer

The **Server Role Designer** window allows you to edit server role for the SQL Server.

**Note:** SQL Azure does not support.

- [Editing Server Role General](#)
- SQL Preview

## Editing SQL Server Server Role General

The **General** tab allows you to set the server role properties which are:

### Role Name

Name of the fixed server role.

### Role Membership

In the grid, check the server role to assign the selected server role to be a member of this server role. Multiple roles can be granted.

## SQL Server Database User Designer

The **Database User Designer** window allows you to set different properties and permissions for the SQL Server database user.

- [Editing Database User General](#)
- [Setting Roles](#)
- [Setting Owned Schemas](#)
- [Setting Database Permissions](#)
- [Setting Object Permissions](#)
- SQL Preview

## Editing SQL Server Database User General

The **General** tab allows you to set database user properties which are:

### User Name

Set name of the database user.

### User Type

Select the type for this database user.

### For Login

Specifies the SQL Server login for which the database user is being created.

#### Login Name

Assign SQL Server login that this database user uses. When this SQL Server login enters the database, it will retrieve the information of this database user.

#### Default Schema

You can specify the first schema that will be searched by the server for this database user.

**Note:** Support from SQL Server 2005 or later and SQL Azure.

### For Certificate

Specifies the certificate for which the database user is being created.

**Note:** Support from SQL Server 2005 or later.

#### Certificate Name

Specify the certificate for this database user.

### For Asymmetric Key

Specifies the asymmetric key for which the database user is being created.

**Note:** Support from SQL Server 2005 or later.

#### Asymmetric Key Name

Specify the asymmetric key for this database user.

## **Without Login**

Specifies this database user not be mapped to an existing login.

**Note:** Support from SQL Server 2005 or later and SQL Azure.

## **Default Schema**

You can specify the first schema that will be searched by the server for this database user.

**Note:** Support from SQL Server 2005 or later and SQL Azure.

## Setting SQL Server Database User Roles

In the grid, check the role to assign this database user to be a member of selected database role. Multiple roles can be granted.

Every database user belongs to the **public** database role. When a user has not been granted or denied specific permissions on a securable, the user inherits the permissions granted to **public** on that securable.

## Setting SQL Server Database User Owned Schemas

You can check the schema listed in **Owned Schemas** tab to change schema ownership to this database user.

**Note:** Support from SQL Server 2005 or later and SQL Azure.


## Setting SQL Server Database User Database Permissions

In the grid, check **Grant**, **With Grant Option** or **Deny** against the database permission listed in **Permission** to assign this database user to have that permission on the database. Multiple permissions can be granted.

To grant (select) or revoke (unselect) all privileges, right-click the grid and select **Grant All**, **Grant All with Grant Option**, **Deny All** or **Revoke All** option.




## Setting SQL Server Database User Object Permissions

To edit the specific object permission of this database user, click  **Add Permission** to open the window and follow the steps below:

- (1) Expand the node in the tree view until reaching to the target object.
- (2) Check the object to show the grid on the right panel.
- (3) In the grid, check **Grant**, **With Grant Option** or **Deny** against the permission listed in **Privilege** to assign this database user to have that permission. Multiple permissions can be granted.

To grant (select) or revoke (unselect) all permissions, right-click the grid and select **Grant All**, **Grant All With Grant Option**, **Deny All** or **Revoke All** option.

**Note:** Click  **Save** to apply any changes you have made.

## SQL Server Database Role Designer

The **Database Role Designer** window allows you to set different properties and permissions for a SQL Server database role.

- [Editing Database Role General](#)
- [Setting Owned Schemas](#)
- [Setting Object Permissions](#)
- SQL Preview

## Editing SQL Server Database Role General

The **General** tab allows you to set database role properties which are:

### Role name

Set name of the database role.

### Owner

You can enter the owner for this database role. This owner can be database user or database role. If the owner is not specify, this database role will be owned by the user who executes the CREATE ROLE.

**Note:** Support from SQL Server 2005 or later and SQL Azure.

### Role Membership


In the grid, check the database user to assign the selected database user to be a member of this database role. Multiple roles can be granted.

## Setting SQL Server Database Role Owned Schemas

You can check the schema listed in **Owned Schemas** tab to change schema ownership to this database role .

**Note:** Support from SQL Server 2005 or later and SQL Azure.

## Setting SQL Server Database Role Object Permissions

To edit the specific object permission of the database role, click  **Add Permission** to open the window and follow the steps below:

- (1) Expand the node in the tree view until reaching to the target object.
- (2) Check the object to show the grid on the right panel.
- (3) In the grid, check **Grant**, **With Grant Option** or **Deny** against the permission listed in **Privilege** to assign this database role to have that permission. Multiple permissions can be granted.

To grant (select) or revoke (unselect) all permissions, right-click the grid and select **Grant All**, **Grant All With Grant Option**, **Deny All** or **Revoke All** option.

**Note:** Click  **Save** to apply any changes you have made.

## SQL Server Application Role Designer

The **Application Role Designer** window allows you to set different properties and permissions for a SQL Server application role.

**Note:** SQL Azure does not support.

- [Editing Application Role General](#)
- [Setting Owned Schemas](#)
- [Setting Object Permissions](#)
- SQL Preview

## Editing SQL Server Application Role General

The **General** tab allows you to set application role properties which are:

### **Role name**

Set name of the application role.

### **Default Schema**

You can specify the first schema that will be searched by the server for this application role.

**Note:** Support from SQL Server 2005 or later and SQL Azure.

### **Password**

Set role's password.

### **Confirm Password**

Re-type the password here.


## Setting SQL Server Application Role Owned Schemas

You can check the schema listed in **Owned Schemas** tab to change schema ownership to this application role .

**Note:** Support from SQL Server 2005 or later and SQL Azure.




## Setting SQL Server Application Role Object Permissions

To edit the specific object permission of the application role, click  **Add Permission** to open the window and follow the steps below:


- (1) Expand the node in the tree view until reaching to the target object.
- (2) Check the object to show the grid on the right panel.
- (3) In the grid, check **Grant**, **With Grant Option** or **Deny** against the permission listed in **Privilege** to assign this application role to have that permission. Multiple permissions can be granted.

To grant (select) or revoke (unselect) all permissions, right-click the grid and select **Grant All**, **Grant All With Grant Option**, **Deny All** or **Revoke All** option.

**Note:** Click  **Save** to apply any changes you have made.

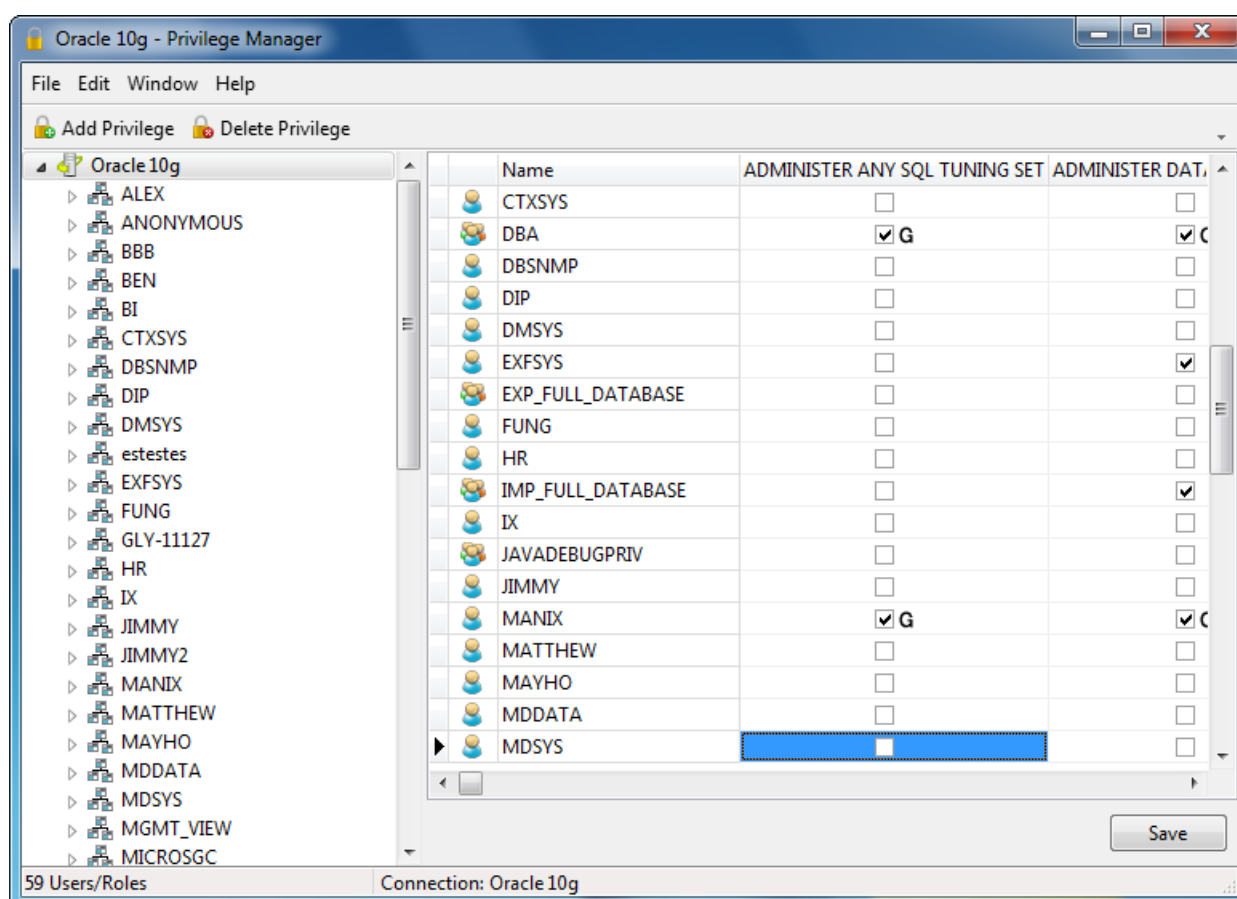
## Privilege Manager

The **Privilege Manager** provides another view on privileges in server and its database objects.

To add privilege, click  **Add Privilege** to open the window and follow the steps below:

- (1) Expand the node in the tree view until reaching to the target object.
- (2) Check the object to show the grid on the right panel.
- (3) In the grid, check the relevant privilege against the user/role listed in **Name** to assign the selected user/role to have that object privilege. Multiple privileges can be granted. You can click on the checkbox to have more choices on the permission setting.

**Note:** Click **Save** to apply any changes you have made.



## Database Maintenance Tasks

Navicat provides a complete solution for maintaining databases and their database objects in MySQL, Oracle, PostgreSQL, SQLite and SQL Server.

- [MySQL Maintenance Tasks](#)
- [Oracle Maintenance Tasks](#)
- [PostgreSQL Maintenance Tasks](#)
- [SQLite Maintenance Tasks](#)
- [SQL Server Maintenance Tasks](#)
- [Server Monitor](#)

## MySQL Maintenance Tasks

Navicat provides a complete solution for most of the native MySQL services, which are intended for database and table maintenance. To make your work with the server easier, Navicat also provides some graphical tools for working with the server as a whole.

### [Analyze Tables](#)

Analyzes and stores the key distribution for the table.

### [Check Tables](#)

Checks the database tables for errors.

### [Optimize Tables](#)

Reclaims the unused space in tables and defragments the data files.

### [Repair Tables](#)

Repairs database tables that are corrupted.

### [Flush](#)

Clears the internal MySQL caches.

## Analyze MySQL Tables

**Analyze Table** analyzes and stores the key distribution for the table. During the analysis, the table is locked with a read lock for MyISAM and BDB. For InnoDB the table is locked with a write lock. Currently, MySQL supports analyzing only for MyISAM, BDB, and InnoDB tables. For MyISAM tables, this statement is equivalent to using `myisamchk --analyze`.

**Hint:** Just simply right-click the table and select **Maintain -> Analyze Tables....**

MySQL uses the stored key distribution to decide in which order tables should be joined when one does a join on something else than a constant.

Analyze Table returns a result set with the following columns:

Column	Value
Table	The table name.
Op	Always analyze.
Msg_type	One of status, error, info, or warning.
Msg_text	The message.

You can check the stored key distribution with the *SHOW INDEX* statement. If the table has not changed since the last **Analyze Table** statement, the table is not analyzed again.

## Check MySQL Tables

**Check Table** checks a table or tables for errors. Currently, MySQL supports checking only for MyISAM, InnoDB and ARCHIVE tables. For MyISAM tables, the key statistics are updated as well.

**Hint:** Just simply right-click the table and select **Maintain -> Check Tables -> desired option**.

Check Table returns a result set with the following columns:

Column	Value
Table	The table name.
Op	Always check.
Msg_type	One of status, error, info, or warning.
Msg_text	The message.

You might get many rows of information for each checked table. The last row has a *Msg\_type* value of *status* and the *Msg\_text* normally should be *OK*. If you do not get *OK*, or *Table is already up to date* you should normally run a repair of the table. *Table is already up to date* means that the storage engine for the table indicated that there was no need to check the table.

The other check options that can be given are shown in the following table:

Type	Meaning
QUICK	Don't scan the rows to check for wrong links.
FAST	Don't scan the rows to check for wrong links.
CHANGED	Only check tables which have been changed since last check or haven't been closed properly.
EXTENDED	Do a full key lookup for all keys for each row. This ensures that the table is 100 % consistent, but will take a long time!

## Optimize MySQL Tables

The main reason for optimizing your table is to reclaim unused space and to defragment the data file. You should optimize a table if you have deleted a large part of a table or if you have made many changes to a table with variable-length rows (tables that have VARCHAR, BLOB, or TEXT columns). Deleted records are maintained in a linked list and subsequent INSERT operations reuse old row positions.

**Hint:** Just simply right-click the table and select **Maintain -> Optimize Tables....**

Currently, MySQL supports optimizing only for MyISAM, InnoDB and BDB tables.

For MyISAM tables, **Optimize Table** works as follows:

1. If the table has deleted or split rows, repair the table.
2. If the index pages are not sorted, sort them.
3. If the table's statistics are not up to date (and the repair could not be accomplished by sorting the index), update them.

## Repair MySQL Tables

**Repair Table** repairs a possibly corrupted table.

**Hint:** Just simply right-click the table and select **Maintain -> Repair Tables -> desired option**.

Repair Table returns a result set with the following columns:

Column	Value
Table	The table name.
Op	Always analyze.
Msg_type	One of status, error, info, or warning.
Msg_text	The message.

You might get many rows of information for each repaired table. The last row has a *Msg\_type* value of *status* and *Msg\_text* normally should be *OK*. If you do not get *OK*, you should try repairing the table with *myisamchk --safe-recover*. Repair Table does not yet implement all the options of *myisamchk*. With *myisamchk --safe-recover*, you can also use options that Repair Table does not support, such as *--max-record-length*.

If **Quick** is given, Repair Table tries to repair only the index tree.

If you use **Extended**, MySQL creates the index row by row instead of creating one index at a time with sorting.



## Flush MySQL

**Flush** clears or reloads various internal caches used by MySQL. To execute Flush, you must have the *Reload* privilege (see MySQL Security Management).

**Hint:** Just simply right-click the connection and select **Flush**.

The following table illustrates the use of **Flush**:

- **Privileges**  
Reloads the privileges from the grant tables in the MySQL database.
- **Hosts**  
Empties the host cache tables. You should flush the host tables if some of your hosts change IP number or if you get the error message *Host 'host\_name' is blocked*. When more than *max\_connect\_errors* errors occur in a row for a given host while connection to MySQL server, MySQL assumes something is wrong and blocks the host from further connection requests. Flushing the host tables allow the host to attempt to connect again.
- **Logs**  
Closes and reopens all log files. If you have specified the update log file or a binary log file without an extension, the extension number of the log file will be incremented by one relative to the previous file. If you have used an extension in the file name, MySQL will close and reopen the update log file.
- **Status**  
Resets most status variables to zero. This is something one should only use when debugging a query.
- **Tables**  
Closes all open tables and forces all tables in use to be closed.

## Oracle Maintenance Tasks

Navicat provides a complete solution for database object maintenance. To make your work with the server easier, Navicat provides some graphical tools for working with the server as a whole.

- [Tables](#)
- [Views](#)
- [Functions/Procedures](#)
- [Indexes](#)
- [Java](#)
- [Materialized Views](#)
- [Materialized View Logs](#)
- [Packages](#)
- [Triggers](#)
- [Types](#)
- [XML Schema](#)
- [Tablespaces](#)

## Oracle Table Maintenance Tasks

Select the table for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### [Enable / Disable Table Lock](#)

Allows / Prevents DDL operations on the table.

### [Enable / Disable Row Movement](#)

Allows / Prevents the database to move a row.

### [Shrink Space](#)

Shrinks space in a table.

### [Move](#)

Relocates data of a nonpartitioned table or of a partition of a partitioned table into a new segment.

### [Collect Statistics](#)

Collects table statistics.

### [Validate Structure](#)

Verifies the integrity of the structure of a table.

## Enable / Disable Table Lock

**Table Lock** locks a table to prevent DDL operations. Oracle Database permits DDL operations on a table only if the table can be locked during the operation. Such table locks are not required during DML operations.

### Enable Table Lock

Choose Enable Table Lock to enable table locks, thereby allowing DDL operations on the table. All currently executing transactions must commit or roll back before Oracle Database enables the table lock.

### Disable Table Lock

Choose Disable Table Lock to disable table locks, thereby preventing DDL operations on the table.

## Enable / Disable Row Movement

**Row Movement** is the moving of rows in tables. It is possible for a row to move, for example, during table compression or an update operation on partitioned data.

### Enable Row Movement

Choose Enable Row Movement to allow the database to move a row, thus changing the rowid.

### Disable Row Movement

Choose Disable Row Movement if you want to prevent the database from moving a row, thus preventing a change of rowid.

## Shrink Space

**Shrink Space** is to compact the table segment. This clause is valid only for segments in tablespaces with automatic segment management. By default, Oracle Database compacts the segment, adjusts the high water mark, and releases the recuperated space immediately.

Compacting the segment requires row movement. Therefore, you must enable row movement for the table you want to shrink before shrink space. Further, if your application has any rowid-based triggers, you should disable them before issuing this clause.

## Move

**Move** relocates data of a nonpartitioned table or of a partition of a partitioned table into a new segment, optionally in a different tablespace, and optionally modify any of its storage attributes.

## Collect Statistics

**Collect Statistics** analyzes the contents of tables. When you analyze a table, the database collects statistics about expressions occurring in any function-based indexes as well. Therefore, be sure to create function-based indexes on the table before analyzing the table.

Oracle Database collects the following statistics for a table. Statistics marked with an asterisk (\*) are always computed exactly.

NUM_ROWS	Number of rows.
* BLOCKS	Number of data blocks below the high water mark - the number of data blocks that have been formatted to receive data, regardless whether they currently contain data or are empty.
* EMPTY_BLOCKS	Number of data blocks allocated to the table that have never been used.
AVG_SPACE	Average available free space in each data block in bytes.
CHAIN_COUNT	Number of chained rows.
AVG_ROW_LEN	Average row length, including the row overhead, in bytes.



## **Validate Structure**

**Validate Structure** verifies the integrity of the structure of a table. The statistics collected by this clause are not used by the Oracle Database optimizer. If the structure is valid, no error is returned. However, if the structure is corrupt, an error message will be shown.

For a table, Oracle Database verifies the integrity of each of the data blocks and rows.

## Oracle View Maintenance Tasks

Select the view for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### Compile

To recompile the view specification or body.

## Oracle Function/Procedure Maintenance Tasks

Select the function/procedure for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### Compile

To recompile the function/procedure specification or body.

### Compile for Debug

To recompile the function/procedure specification or body and instruct the PL/SQL compiler to generate and store the code for use by the PL/SQL debugger.

## Oracle Index Maintenance Tasks

Select the index for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### Rebuild

To re-create an existing index or one of its partitions or subpartitions. If the index is marked unusable, then a successful rebuild will mark it usable.

### Make Unusable

To make the index unusable. An unusable index must be rebuilt, or dropped and re-created, before it can be used.

### Coalesce

To instruct Oracle Database to merge the contents of index blocks where possible to free blocks for reuse.

### Compute Statistics

To compute the statistics of the index.

### Monitoring Usage

To begin monitoring the index. Oracle Database first clears existing information on index use, and then monitors the index for use until choosing No Monitoring Usage.

### No Monitoring Usage

To terminate monitoring of the index.

## Oracle Java Maintenance Tasks

Select the Java for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### **Compile or Resolve**

To resolve the primary Java class schema object.

### **Set AuthID Current User**

Set the invoker rights to AUTHID CURRENT\_USER.

### **Set AuthID Definer**

Set the invoker rights to AUTHID DEFINER.

## Oracle Materialized View Maintenance Tasks

Select the materialized view for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### Enable Row Movement

To enable row movement.

### Shrink

To compact the materialized view segment. By default, Oracle Database compacts the segment, adjusts the high water mark, and releases the recuperated space immediately.

### Compile

To explicitly revalidate a materialized view. If an object upon which the materialized view depends is dropped or altered, then the materialized view remains accessible, but it is invalid for query rewrite. You can choose this option to explicitly revalidate the materialized view to make it eligible for query rewrite.

### Force Refresh

To perform a refresh.

## Oracle Materialized View Log Maintenance Tasks

Navicat provides a complete solution for materialized view log maintenance.

Select the materialized view log for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### Enable Row Movement

To enable row movement. Row movement indicates that rowids will change after the flashback occurs.

### Disable Row Movement

To disable row movement.

### Shrink Space

To compact the materialized view log segments. By default, Oracle Database compacts the segment, adjusts the high water mark, and releases the recuperated space immediately.

## Oracle Package Maintenance Tasks

Select the package for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### Compile

To recompile the package specification or body.

### Compile Debug

To recompile the package specification or body and instruct the PL/SQL compiler to generate and store the code for use by the PL/SQL debugger.



## Oracle Trigger Maintenance Tasks

Select the trigger for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### **Enable**

To enable the trigger.

### **Disable**

To disable the trigger.

### **Compile**

To explicitly compile the trigger, whether it is valid or invalid. Explicit recompilation eliminates the need for implicit run-time recompilation and prevents associated run-time compilation errors and performance overhead.

### **Compile for Debug**

To recompile the trigger and instruct the PL/SQL compiler to generate and store the code for use by the PL/SQL debugger.

## Oracle Type Maintenance Tasks

Select the type for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### Compile

To compile the type specification and body.

### Compile Debug

To recompile the type specification or body and instruct the PL/SQL compiler to generate and store the code for use by the PL/SQL debugger.

## Oracle XML Schema Maintenance Tasks

Select the XML Schema for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### Compile

To re-compile an already registered XML schema. This is useful for bringing a schema in an invalid state to a valid state.

### Purge

To remove the XML Schema completely from Oracle XML DB in Oracle 11g.

## Oracle Tablespace Maintenance Tasks

Select the tablespace for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### **[Read Only]**

To place the tablespace in transition read-only mode. In this state, existing transactions can complete (commit or roll back), but no further DML operations are allowed to the tablespace except for rollback of existing transactions that previously modified blocks in the tablespace.

### **Read Write**

To indicate that write operations are allowed on a previously read-only tablespace.

### **Online**

To take the tablespace online.

### **Offline**

To take the tablespace offline.

### **Normal**

To flush all blocks in all datafiles in the tablespace out of the system global area (SGA).

### **Temporary**

Oracle Database performs a checkpoint for all online datafiles in the tablespace but does not ensure that all files can be written.

### **Immediate**

Oracle Database does not ensure that tablespace files are available and does not perform a checkpoint.

### **Coalesce**

To combine all contiguous free extents into larger contiguous extents for each datafile in the tablespace.

### **Shrink Space**

To reduce the amount of space the tablespace is taking. This is valid only for temporary tablespaces in Oracle 11g.

## PostgreSQL Maintenance Tasks

Navicat provides a complete solution for most of the native PostgreSQL services, which are intended for database and table maintenance. To make your work with the server easier, Navicat also provides some graphical tools for working with the server as a whole.

### [Analyze](#)

Collects statistics about the contents of tables in the database.

### [Vacuum](#)

Reclaims storage occupied by deleted tuples.

### [Reindex](#)

Rebuilds an index using the data stored in the index's table.

## Analyze PostgreSQL Database and Tables

**Analyze** collects statistics about the contents of tables in the database, and stores the results in the system table *pg\_statistic*. Subsequently, the query planner uses these statistics to help determine the most efficient execution plans for queries.

### Analyze Database

Just simply right-click the database and select **Maintain -> Analyze Database....**

### Analyze Table

Just simply right-click the table and select **Maintain -> Analyze Tables....**

Parameters	
Verbose	Enables display of progress messages. (Default enabled in Navicat)

Outputs	
When Verbose is specified, Analyze emits progress messages to indicate which table is currently being processed. Various statistics about the tables are printed as well.	

## Vacuum PostgreSQL Database and Tables

**Vacuum** reclaims storage occupied by deleted tuples. In normal PostgreSQL operation, tuples that are deleted or obsoleted by an update are not physically removed from their table; they remain present until a Vacuum is done. Therefore it's necessary to do Vacuum periodically, especially on frequently-updated tables.

### Vacuum Database

Just simply right-click the database and select **Maintain -> Vacuum Database -> desired option**.

### Vacuum Table

Just simply right-click the table and select **Maintain -> Vacuum Tables -> desired option**.

Parameters	
Full	Selects "full" vacuum, which may reclaim more space, but takes much longer and exclusively locks the table.
Freeze	Selects aggressive "freezing" of tuples.
Analyze	Updates statistics used by the planner to determine the most efficient way to execute a query.
Verbose	Prints a detailed vacuum activity report for each table. (Default enabled in Navicat)

Outputs
When Verbose is specified, Vacuum emits progress messages to indicate which table is currently being processed. Various statistics about the tables are printed as well.

## Reindex PostgreSQL Database and Tables

**Reindex** rebuilds an index using the data stored in the index's table, replacing the old copy of the index. There are several scenarios in which to use Reindex:

- An index has become corrupted, and no longer contains valid data.
- An index has become "bloated", that it contains many empty or nearly-empty pages.
- You have altered a storage parameter (such as fill factor) for an index, and wish to ensure that the change has taken full effect.
- An index build with the CONCURRENTLY option failed, leaving an "invalid" index.

### Reindex Database

Just simply right-click the database and select **Maintain -> Reindex Database....**

### Reindex Table

Just simply right-click the table and select **Maintain -> Reindex Tables....**



## SQLite Maintenance Tasks

Navicat provides a complete solution for most of the SQLite services, which are intended for database, table and index maintenance. To make your work with the server easier, Navicat also provides some graphical tools for working with the server as a whole.

- [Databases and Tables](#)
- [Indexes](#)

## SQLite Database and Table Maintenance Tasks

To perform the following tasks, right-click and select the database or table for maintaining in the object pane.

### [Analyze](#)

Collects statistics about the contents of indexes.

### [Vacuum](#)

Cleans empty spaces from the database.

### [Reindex](#)

Rebuilds an index using the data stored in the index's table.

### [View Master Table](#)

Views all tables and indexes located in database.

## Analyze SQLite Database and Tables

**Analyze** collects statistics about the indexes and stores the results in a special table in the database to help make better index selection. Subsequently, the query optimizer uses these statistics to help make better index choices.

### Analyze Database

Just simply right-click the database and select **Maintain -> Analyze Database....**

### Analyze Table

Just simply right-click the table and select **Maintain -> Analyze Tables....**

Parameters	
Database	Analyzes all indexes in one database when the database name is specified.
Table	Analyzes all indexes in one table when the table name is specified.

## **Vacuum SQLite Database**

**Vacuum** cleans the empty spaces remained by dropping objects frequently from the database. It cleans the main database by copying its contents to a temporary database file and reloading the original database file from the copy. It only works on the main database. It is not possible to vacuum an attached database file.

### **Vacuum Database**

Just simply right-click the database and select **Maintain -> Vacuum Database...**

## Reindex SQLite Database and Tables

**Reindex** rebuilds an index using the data stored in the index's table, replacing the old copy of the index. Reindex is used when the definition of a collation sequence has changed.

### Reindex Database

Just simply right-click the database and select **Maintain -> Reindex Database....**

### Reindex Table

Just simply right-click the table and select **Maintain -> Reindex Tables....**

## View Master Table

**View Master Table** allows viewing all tables and indexes located in an SQLite database. Every SQLite database has a special table named *SQLITE\_MASTER* table that defines the schema for the database.

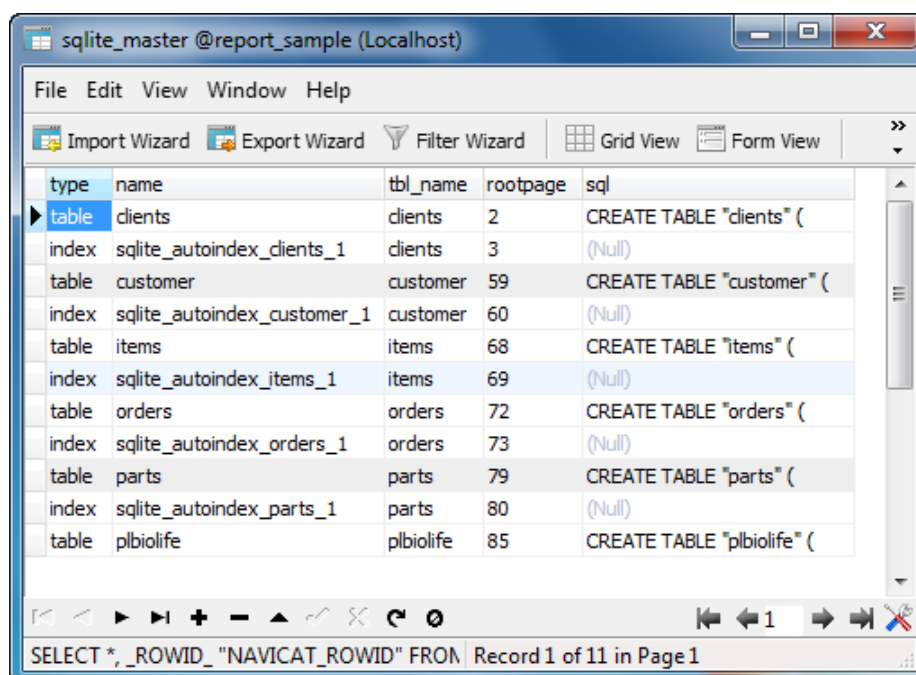
For tables, the **type** field will always be **table** and the **name** field will be the name of the table. For indexes, **type** is equal to **index**, **name** is the name of the index and **tbl\_name** is the name of the table to which the index belongs. For both tables and indexes, the **sql** field is the text of the original *CREATE TABLE* or *CREATE INDEX* statement that created the table or index. For automatically created indexes (used to implement the PRIMARY KEY or UNIQUE constraints) the **sql** field is NULL.

The *SQLITE\_MASTER* table is read-only. You cannot change this table using UPDATE, INSERT, or DELETE. The table is automatically updated by *CREATE TABLE*, *CREATE INDEX*, *DROP TABLE*, and *DROP INDEX* commands.

Temporary tables do not appear in the *SQLITE\_MASTER* table. Temporary tables and their indexes and triggers occur in another special table named *SQLITE\_TEMP\_MASTER*. *SQLITE\_TEMP\_MASTER* works just like *SQLITE\_MASTER* except that it is only visible to the application that created the temporary tables.

## View Master Table

Just simply right-click the database and select **View Master Table...**



## SQLite Index Maintenance Tasks

Select the index for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### Reindex

To delete and recreate the index from scratch. This is useful when the definition of a collation sequence has changed.

## SQL Server Maintenance Tasks

Navicat provides a complete solution for most of the SQL Server services, which are intended for assembly, index and trigger maintenance. To make your work with the server easier, Navicat also provides some graphical tools for working with the server as a whole.

- [SQL Azure Firewall Rules](#)
- [Assembly](#)
- [Indexes](#)
- [Triggers](#)



## SQL Azure Firewall Rules (Available only for SQL Azure)

You cannot connect to SQL Azure until you have granted your client IP access. To access SQL Azure database from your computer, ensure that your firewall allows outgoing TCP communication on TCP port 1433. You must have at least one firewall rule before you can connection to SQL Azure.

Right-click the SQL Azure connection and select the **SQL Azure Firewall Rules...** from the popup menu. You can add new rule by providing a range of IP address.

## SQL Server Assembly Maintenance Tasks

Select the assembly for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### **Set Visible**

To show the assembly.

### **Set Invisible**

To hide the assembly.

## SQL Server Index Maintenance Tasks

Select the index for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.

### **Rebuild**

To rebuild and enable the index.

### **Reorganize**

To reorganize the enabled index.

### **Disable**

To disable the index.

## SQL Server Trigger Maintenance Tasks

Select the server trigger, database trigger or trigger for maintaining in the object pane. Right-click and select the **Maintain** from the popup menu.


### Enable

To enable the trigger.

### Disable

To disable the trigger.

## **Server Monitor (Available only in Full version & only for MySQL, Oracle, PostgreSQL and SQL Server)**

Navicat provides **Server Monitor** to view the properties of the selected server. Just simply choose **Tools** ->  **Server Monitor** and select the target server type from the main menu.

**Note:** SQL Azure does not support.

- [Process List](#)
- [Variables](#)
- [Status](#)

## Process List

**Process List** tab displays a list of processes from all the database servers selected in the check-list box.

To stop the selected process, just simply click the  **End Process** button.

**Hint:** The process list cannot be edited.

### Auto refresh every      seconds

If you want to take action on auto-refreshing the server in assigned seconds, just simply choose **View -> Set Auto Refresh Time** and enter an auto refresh value. To disable auto refresh feature, choose **View -> Auto Refresh**.

**Note:** Effect will take once you assign the value.

The process list provides the following information:

- Server name that is given while setting the connection.
- Process ID on the server.
- Serial number of the process. (Available only for Oracle)
- Current user who log in to the server.
- Host from which the user is connected.
- Database that the user is currently used. (Available only for MySQL, PostgreSQL and SQL Server)
- Last command that was issued by the user.
- Time, state and info of the process. (Available only for MySQL, Oracle and PostgreSQL)
- CPU Time and state of the process. (Available only for SQL Server)


## Variables (Available only for MySQL, Oracle and PostgreSQL)

**Variables** tab displays the list of all server variables and their values. The variables list is retrieved from the server(s) by issuing the SQL statement.

For MySQL Server: *SHOW VARIABLES*.

For Oracle Server: *SHOW ALL*.

For PostgreSQL Server: *SHOW ALL*.

**Hint:** To edit variable value in MySQL and Oracle servers, just simply click  or press Ctrl+Enter to open the editor for editing.

**Hint:** The value in PostgreSQL server cannot be edited here. (Those variables can be set using the *SET* statement, by editing the *postgresql.conf* configuration file.)

## Status (Available only for MySQL, Oracle and PostgreSQL)

**Status** tab displays the list of all server status of the server. The status list is retrieved from the MySQL server(s) by issuing the *SHOW STATUS* statement.

**Hint:** The status cannot be edited here.



## Options

Navicat provides a complete user interface customization for various options of all tools.

Just simply click **Tools** -> **Options** from the main menu.

- [General Options](#)
- [Appearance Options](#)
- [Model Options](#)
- [Miscellaneous Options](#)

## General Options

### General

#### ☒ **Windows in taskbar**

Every new window that is opened automatically is shown on the Windows Taskbar. With this option is disabled, all instances (e.g. tables, queries) will be closed while main Navicat exits.

**Hint:** Reopen Navicat to take effect.

#### ☒ **Allow Multiple Form Instances**

With this option is on, you allow opening multiple instances of the same selected window.

#### ☒ **Allow Multiple Navicat Instances**

Unchecking this item means that clicking on the Navicat shortcut will re-activate the running instance of Navicat and not launch a new copy.

#### ☒ **Click to refresh**

Refreshes the object pane list whenever you click on the objects.

#### ☒ **Show function wizard**

Displays the function wizard (MySQL, Oracle, PostgreSQL or SQL Server) when you create a new function/procedure.

#### ☒ **Ask to save new queries/profiles before closing**

With this option is on, Navicat will prompt you to save new queries or profiles every time when you quit the relevant sub-window.

## Docking Options

### Docking

#### ☒ **Using docking**

Allows you to define the basic window behavior style - Docking Windows (like Macromedia Dreamweaver) or Floating Windows (like Borland IDE).

#### ☒ **Dock opened windows**

Chooses either dock the opened windows to **main window** or **dock window**.

#### ☒ **Follow the last docking style**

Inherits the latest docking style.

## Code Insight Options (Available only in Full Version)

### Code Completion

#### ☒ Use Code Completion

When you type the . (dot) symbol between the object names, SQL Editor will offer you a popup list that showing some variants for the code completion, see Code Completion.

#### Delay

You can change the time the popup list takes to appear.

### Word Completion

#### ☒ Use Word Completion

When you type the first character of words, SQL Editor will offer you a popup list that showing some variants for the word completion.

#### Delay

You can change the time the popup list takes to appear.

### Syntax Highlighting

#### ☒ Use Syntax Highlighting

Syntax highlight helps viewing codes clearly. Codes are highlighted in SQL Editor with different colors and fonts according to the categories they belong to. The syntax highlighting feature can be limited by setting the maximum file size (e.g. 10) in **Apply Syntax Highlighting for statement size below (MB)** to increase performance.

## Auto Save Options

### Auto Save

#### ☒ Use Auto Save

Saves automatically after modifications in SQL Editor by defining the **Auto Save Interval (s)** (e.g. 30).

## Appearance Options

### General

#### ☒ Show toolbar caption

Shows text on toolbar buttons in sub-windows. Otherwise, only buttons will be presented.



### Font

#### Grid

Defines the **Grid Font** name and **Size** used by table grid.

#### Editor

Defines the **Editor Font** name and **Size** used by editor.

#### Console

Defines the **Console Font** name and **Size** used in console.

## Color Options

### Grid Colors

Defines the background colors of the table grid.

#### ☒ **Use three colors**

Displays the table grid background by using three different colors respectively for viewing data clearly.

### Text Colors

This color settings allows you to format your SQL queries in SQL Editor with colored syntax highlighting for your SQL statements to improve readability.

Sets font colors of the SQL Editor uses to mark out different text fragments: Common, Keywords, Comments, Strings and Numbers. Just simply click on the color boxes and choose your desired color from the **Color-Selection** dialog window.

## Main Window Options

### General

#### ☒ **Show table hint**

While you roll the mouse pointer over a table within the object pane, you could get a popup hint giving details about the table structure.

#### ☒ **Show objects in connection tree**

Displays database/schema objects using the tree structure in navigation pane. To expand node, click the plus sign (+) or double-click the node.

**Hint:** Reopens the database/schema to take effect.

#### ☒ **Use Customized Connection Order**

Checks this option to customize the connection tree order in navigation pane. (using drag and drop method)

#### ☒ **Show system items (PostgreSQL, SQL Server) (Available only for PostgreSQL and SQL Server)**

Checks this option to show all the system objects such as *information\_schema* and *pg\_catalog* schemas.

**Hint:** Reopens the database/schema to take effect.

#### ☒ **Show auto index (SQLite) (Available only for SQLite)**

Checks this option to show auto index generated for SQLite table in Index.



## Editor Options

### Editor

#### ☒ **Show Line Number**

Displays line numbers at the side of the editor for easily reference.

#### ☒ **Use Code Folding**

Code folding allows codes to collapse as a block and only the first line displayed in the editor, see Code Folding.

#### ☒ **Use Brace Highlighting**

Highlights the pair of braces when your cursor moves to either one brace for easily reference, see Brace Highlight.

### Tab Width

Enter the number of characters that a tab occupies, e.g. 5.

## Data/Grid Options

### General

#### ☒ **Show primary key warning**

Checks this option if you require notification while opening the table with no primary key is set.

#### ☒ **Show TEXT Blob fields in data grids**

If this option is on, data which set as TEXT field type is visible in table grid. Otherwise, (WIDEMEMO) will be shown.

#### ☒ **Limit Records**

Checks this option if you want to limit the number of records showed on each page in table grid/foreign key data selection globally. Otherwise, all records will be displayed in one single page.

#### **records per page**

Sets the **records per page** value (e.g. 1000) in the edit field. The number representing the number of records showed per page in table grid.

**Note:** To adjust the settings for particular table, see Table Viewer.

#### **records per page in foreignkey editor**

Sets the **records per page in foreignkey editor** value (e.g. 100) in the edit field. The number representing the number of records showed per page in Foreign Key Data Selection.

#### ☒ **Synchronize current record**

When updating/inserting a row in table grid (if primary key exists), it will reload this record from server.

For example: a table with 3 columns - id, name and timestamp. If you update the name column, the timestamp will update immediately in the grid.

### Row height

Defines the height of the row (e.g. 17) used in editor.



**Note:** To adjust the settings for particular table, see Formatting Table Grid.

## Column Width

Defines the width of the column (e.g. 150) used in editor.

**Note:** To adjust the settings for particular table, see Formatting Table Grid.

### ☒ **Auto commit (Available only for Oracle, SQLite and SQL Server)**

Checks this option if you require auto commit of changing records in table grid. Otherwise, you allow choosing  **Commit** or  **Rollback** to commit or rollback the changes. See Table Viewer.

## Display Format Options

### Display formats

Data of types integer, float, date, time and datetime can be formatted when displayed on data grids. Type the format here to change the format. If the formats are left blank, default format will be used. For date, time and datetime fields, default formats will be the system datetime formats.

Display formats

Integer:

Float:

Date:

Time:

DateTime:

Example:
6/7/2011 14:37:16

Output:
07/06/2011 14:37:16

Formats are defined by constructing a string using these format specifiers:

### Numeric fields

Specifier	Represents
0	Digit placeholder. If the value being formatted has a digit in the position where the "0" appears in the format string, then that digit is copied to the output string. Otherwise, a "0" is stored in that position in the output string. (e.g. with 0000 placed in the Integer field, all the integer output from the table will have 0012 in format)
#	Digit placeholder. If the value being formatted has a digit in the position where the "#" appears in the format string, then that digit is copied to the output string. Otherwise, nothing is stored in that position in the output string. (e.g. with ## ## placed in the Integer field, all the integers output from the table will have 12 34 in format)
.	Decimal point. The first "." character in the format string

	determines the location of the decimal separator in the formatted value; any additional "." characters are ignored. The actual character used as a the decimal separator in the output string is determined by the DecimalSeparator global variable. The default value of DecimalSeparator is specified in the Number Format of the Region and Language Options section in the Windows Control Panel.
,	Thousand separator. If the format string contains one or more "," characters, the output will have thousand separators inserted between each group of three digits to the left of the decimal point. The placement and number of "," characters in the format string does not affect the output, except to indicate that thousand separators are wanted. The actual character used as a the thousand separator in the output is determined by the ThousandSeparator global variable. The default value of ThousandSeparator is specified in the Number Format of the Region and Language Options section in the Windows Control Panel.
E+	Scientific notation. If any of the strings "E+", "E-", "e+", or "e-" are contained in the format string, the number is formatted using scientific notation. A group of up to four "0" characters can immediately follow the "E+", "E-", "e+", or "e-" to determine the minimum number of digits in the exponent. The "E+" and "e+" formats cause a plus sign to be output for positive exponents and a minus sign to be output for negative exponents. The "E-" and "e-" formats output a sign character only for negative exponents.
'xx'/"xx"	Characters enclosed in single or double quotes are output as-is, and do not affect formatting.
;	Separates sections for positive, negative, and zero numbers in the format string.

The locations of the leftmost "0" before the decimal point in the format string and the rightmost "0" after the decimal point in the format string determine the range of digits that are always present in the output string.

The number being formatted is always rounded to as many decimal places as there are digit placeholders ("0" or "#") to the right of the decimal point. If the format contains no decimal point, the value being formatted is rounded to the nearest whole number.

If the number being formatted has more digits to the left of the decimal separator than there are digit placeholders to the left of the "." character in the format string, the extra digits are output before the first digit placeholder.

The following table shows the effect of various format strings:

Display Format	Value	Result	Comment
#.##	12.2	12.2	Note extra digit to left of decimal still appears.
#.00	2.5	2.50	Note extra zero: field will always show two decimal places.
00.##	.006	00.01	Note extra 0s to right of decimal point and rounding to two decimal places.

To allow different formats for positive, negative, and zero values, the format string can contain between one and three sections separated by semicolons.

**One section:** The format string applies to all values.

**Two sections:** The first section applies to positive values and zeros, and the second section applies to negative values.

**Three sections:** The first section applies to positive values, the second applies to negative values, and the third applies to zeros.

If the section for negative values or the section for zero values is empty, that is, if there is nothing between the semicolons that delimit the section, the section for positive values is used instead.

If the section for positive values is empty, or if the entire format string is empty, the value is formatted using general floating-point formatting with 15 significant digits. General floating-point formatting is also used if the value has more than 18 digits to the left of the decimal point and the format string does not specify scientific notation.

## Date Time fields

Specifier	Displays
c	The date using the format given by the ShortDateFormat global variable, followed by the time using the format given by the LongTimeFormat global variable. The time is not displayed if the fractional part of the DateTime value is zero.
d	The day as a number without a leading zero (1-31).
dd	The day as a number with a leading zero (01-31).
ddd	The day as an abbreviation (Sun-Sat) using the strings given by the ShortDayNames global variable.
dddd	The day as a full name (Sunday-Saturday) using the strings given by the LongDayNames global variable.
dddddd	The date using the format given by the ShortDateFormat global variable.
ddddddd	The date using the format given by the LongDateFormat global variable.
m	The month as a number without a leading zero (1-12). If the m specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mm	The month as a number with a leading zero (01-12). If the mm specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
mmm	The month as an abbreviation (Jan-Dec) using the strings given by the ShortMonthNames global variable.
mmmm	The month as a full name (January-December) using the strings given by the LongMonthNames global variable.
yy	The year as a two-digit number (00-99).
yyyy	The year as a four-digit number (0000-9999).
h	The hour without a leading zero (0-23).
hh	The hour with a leading zero (00-23).
n	The minute without a leading zero (0-59).
nn	The minute with a leading zero (00-59).
s	The second without a leading zero (0-59).
ss	The second with a leading zero (00-59).
t	The time using the format given by the ShortTimeFormat

	global variable.
tt	The time using the format given by the LongTimeFormat global variable.
am/pm	The time using the 12-hour clock for the preceding h or hh specifier, followed by "am" for any hour before noon, or "pm" for any hour after noon. The am/pm specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
a/p	The time using the 12-hour clock for the preceding h or hh specifier, followed by "a" for any hour before noon, or "p" for any hour after noon. The a/p specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
ampm	The time using the 12-hour clock for the preceding h or hh specifier, followed by the contents of the TimeAMString global variable for any hour before noon, or the contents of the TimePMString global variable for any hour after noon.
/	The date separator character given by the DateSeparator global variable.
:	The time separator character given by the TimeSeparator global variable.
'xx'/"xx"	Characters enclosed in single or double quotes are displayed as-is, with no formatting changes.

Format specifiers may be written in uppercase or lowercase letters; both produce the same result.



## Model Options (Available only in Full Version)

### General

#### ☒ **Highlight Objects**

With this option is on, when a mouse cursor hovers over an object, Navicat will highlight its border with blue color.

#### ☒ **Highlight with Relation**

With this option is on, when a mouse cursor hovers over a relation, Navicat will highlight it with blue or green color indicating relationships between two tables.

#### ☒ **Guess Field Type**

With this option is on, Navicat will predict field types when you design field without using Table Designer.

## Miscellaneous Options

### Process Priority

Priority indicates the priority used when scheduling the thread. Adjust the priority higher or lower as needed.

### File Path

By default, most of the files are located in Settings Save Path. However, some profiles are stored under the **profiles** directory.

All the log files are stored in the sub-directory called **logs**, you can modify to any paths you prefer.

## File Association Options (Available only in Full Version)

### File Association

File associations are what the Navicat uses to open its saved files with Navicat. For example, a .npt file (Data Transfer profile) will open with Data Transfer windows, and a .npi (Import Wizard profile) will open by default with Import Wizard.

**Note:** In Vista or above, you need to click **File Association** button to open the list. In XP or below, file association is listed in here.

## OCI Options (Available only for Oracle)

### General

#### OCI library (oci.dll)

Choose the **Oracle Client/Oracle Instant Client** folder path that includes the OCI library (oci.dll) for Basic/TNS connection.

**Oracle Instant Client** is the simplest way to deploy a full Oracle Client application built with OCI, OCCI, JDBC-OCI, or ODBC drivers. It provides the necessary Oracle Client libraries in a small set of files. It has already included in Navicat installation folder. You can also download **Oracle Client / Oracle Instant Client** through -

### Downloading

#### Oracle Client

<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>

#### Oracle Instant Client

<http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html>

### Installation Guide

#### Oracle Client

[http://download.oracle.com/docs/cd/B28359\\_01/install.111/b32302/toc.htm](http://download.oracle.com/docs/cd/B28359_01/install.111/b32302/toc.htm)

#### Oracle Instant Client

<http://www.oracle.com/technology/tech/oci/instantclient/index.html>

### SQL\*Plus

By default, Navicat will look for the SQL\*Plus under client folder (e.g. C:\Oracle\product\11.1.0\client\_1\BIN). However, you have to specify the location of the SQL\*Plus if Navicat cannot locate under the default path.

See also:

General Settings for Oracle

Oracle Console

## Useful Tools


Navicat provides variety of tools that improves user experience when using Navicat.

- [ER Diagram](#)
- [Virtual Grouping](#)
- [Connection Colorings](#)
- [Favorites](#)
- [Find in Database](#)
- [Search Filter](#)

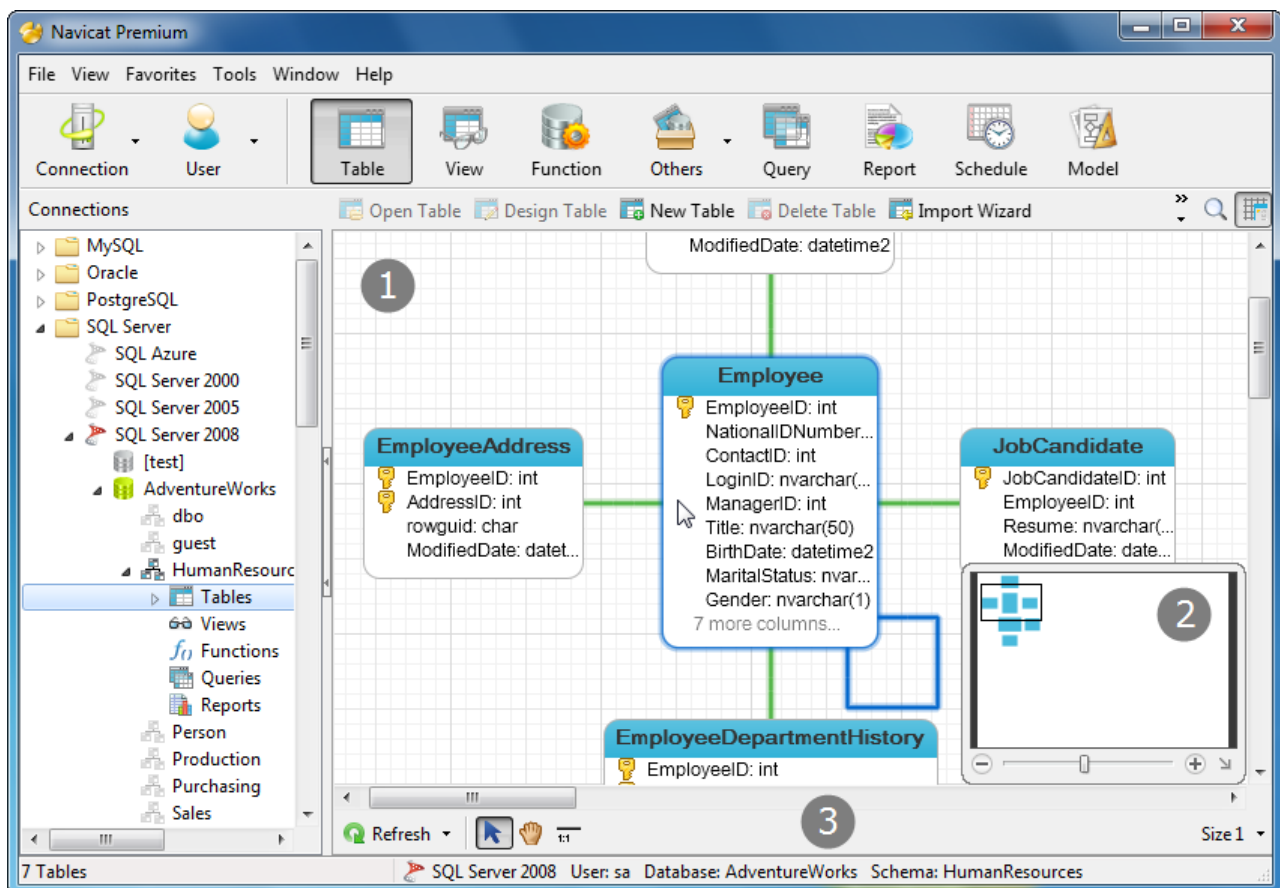
## ER Diagram (Available only in Full Version)

Besides List and Detail views, Navicat enhances table viewing to a new ER Diagram view. In this ER Diagram view, you can view table fields and relationships between tables in a database/schema graphically. It also allows adding foreign key constraints to tables directly.

**Note:** Only tables provide ER Diagram view in Navicat. Other database objects only provide List view and Detail view.

Just simply choose View -> **ER Diagram** in the main menu or click  in the table object pane toolbar. ER diagram will be created automatically if the selected database contains tables.

**Hint:** ER Diagram files are stored under Settings Save Path.



## 1. Object Pane


Displays table fields and relationships between tables in a database.

**Note:** Double-click a table in ER Diagram view will open the Table Designer, while double-click a table in List view and Detail view will open the Table Viewer.

**Note:** The tabs and options in the designer depend on the diagram database type you have chosen. For the settings of different tabs, see Database Object Management.

### Add Relation

To add a relation

- Click  from the bottom toolbar.
- Drag the source table field and drop to the target table field.
- Edit foreign key properties in the Table Designer.

### Edit Relation

To edit a relation

- Select the relation for editing in the object pane.
- Right-click and select the **Design Foreign Key** from the popup menu.
- Edit foreign key properties in the Table Designer.

### Delete Relation

To delete a relation

- Select the relation for deleting in the object pane.
- Right-click and select the **Delete Foreign Key** from the popup menu.
- Confirm deleting in the dialog window.

### Add Vertex to Relation

Select relation in diagram, then press and hold the Shift key. Click on the relation to add vertex.

### Delete Vertex on Relation

Select relation in diagram, then press and hold the Shift key. Click on the vertex.

## 2. Navigator

To zoom in or zoom out the selected area of the diagram, adjust the slider of the Navigator. Same effect can be achieved with keyboard shortcuts:

Zoom In: [Ctrl++ ] or [Ctrl+Mousewheel up]



Zoom out: [Ctrl+-] or [Ctrl+Mousewheel down]

## 3. Toolbar

### **Refresh**

Click to refresh the ER Diagram.

### **Regenerate ER Diagram**

Click the  **Refresh** together with the  down arrow to choose **Regenerate ER Diagram**. It regenerates the ER Diagram with using auto layout feature.

### **Move Diagram**

Click to switch to hand mode. Press and hold the Space key, then move the diagram.

### **New Relation**

Click to create a relation between two table fields.

### **Paper Size**

Select paper size from drop-down list, corresponding paper size will reflect in Navigator.



## Virtual Grouping (Available only in Full Version)

Virtual Group aims to provide a platform for logical grouping objects by categories, so that all objects are effectively preserved.

Virtual Grouping can be applied on Connection, Table, View, Function, Query, Report, Backup and Schedule. Just simply right-click in the navigation pane and select **Manage Group** from the popup menu.

### Create Group

To create a new group

- Select the panel (navigation pane/object pane) that you wish to apply the virtual grouping.
- Right-click and select **New Group** or **Manage Group** -> **New Group** from the popup menu.
- Define the group name.
- Right-click the object(s) (e.g. connection, table, view etc) and select **Manage Group** -> **Move To** from the popup menu or just simply drag and drop the object(s) into the newly created group.

### Edit Group

To change the name of a group

- Select the group for editing in the navigation pane.
- Right-click and select the **Rename** from the popup menu.

### Handle Group

To move the object(s) back to the top-level

- Select the object(s) for moving in the navigation pane.
- Right-click the object(s) and select **Manage Group** -> **Exclude From Group** from the popup menu or just simply drag and drop the object(s).

## Hidden Groups

To hide connection groups

- Choose **View -> Flatten Connection Tree**.

To hide object groups

- Choose **View -> Flatten Object List**.

## Delete Group

To delete a group

- Select the group for deleting in the navigation pane.
- Right-click and select **Delete Group** from the popup menu.
- Confirm deleting in the dialog window.

## Connection Colorings

Navicat provides highlighting connections by colors for identifying connections and their database objects. The highlighted color displays in connection name in navigation pane and menu bar in its database object window.

To highlight a connection, simply right-click the connection, select **Color** and your preferred color from the popup menu.

## Favorites (Available only in Full Version)

Navicat Favorites are links to tables/views/other database objects that you visit frequently. By adding a path to your favorites list, you can go to that tables/views/other database objects with a single click, instead of having to navigate the connection and databases in the navigation pane.

### Add Favorite

To add a link to the favorites list:

1. Open the database object.
2. Choose **File** -> **Add To Favorites...** or press Shift+Ctrl+#.
3. Enter the **Favorite Name** and select the **Favorite ID**.

### Open Favorite

To open a database object from the favorites list:

1. Choose **Favorites** -> **favorite\_name** in Navicat main window or press Ctrl+#.

### Delete Favorite

To remove a link from the favorites list:

1. Choose **Favorites** -> **Clear Favorites** -> **favorite\_name**.

To remove all links from the favorites list:

1. Choose **Favorites** -> **Clear Favorites** -> **Clear All**.

**Note:** # represents 0, 1, 2, 3, 4, 5, 6, 7, 8 or 9.


## Find in Database (Available only in Full Version)

Navicat provides a **Find in Database** feature offers searching table records within a database.

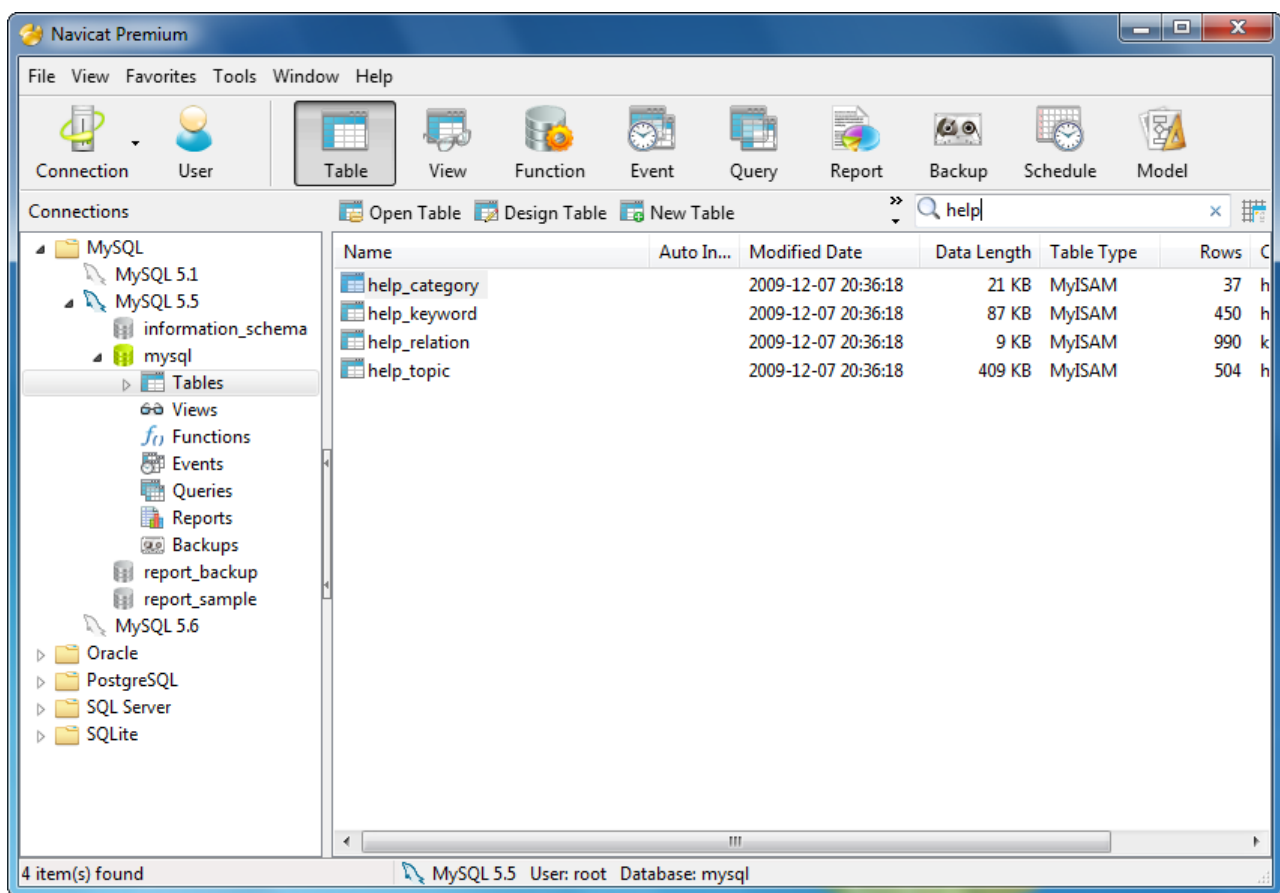
- Select the database/schema.
- Right-click and select **Find in Database.../Find in Schema...** from the popup menu.
- Enter keyword and select the search criteria.
- Double-click the table in Find Results to view the record.

## Search Filter

**Search Filter** allows you to filter the object names that contain the filter string in object pane.

Search Filter can be applied on Table, View, Function, Query, Report, Backup, Schedule and other database objects. Just simply click  and specify a filter string in the Navicat main window.

To remove the filter, simply delete the filter string.



## Navicat Commands (Available only in Full Version)

### Navicat Premium

Start Navicat from command line:

Navicat Objects	Server Type	Command Lines	File Extensions
Backup	MySQL, PostgreSQL and SQLite	Navicat.exe /backup mysql ConnectionName DatabaseName	compressed (.psc), uncompressed (.psb)
		Navicat.exe /backup pgsql ConnectionName DatabaseName SchemaName	
		Navicat.exe /backup sqlite ConnectionName DatabaseName ProfileName	
Backup Server	MySQL, PostgreSQL and SQLite	Navicat.exe /backupserver mysql ConnectionName	
		Navicat.exe /backupserver pgsql ConnectionName	
		Navicat.exe /backupserver sqlite ConnectionName	
Backup Database	MySQL, PostgreSQL and SQLite	Navicat.exe /backupdatabase mysql ConnectionName DatabaseName	
		Navicat.exe /backupdatabase pgsql ConnectionName DatabaseName	
		Navicat.exe /backupdatabase sqlite ConnectionName DatabaseName	
Backup Profile	MySQL, PostgreSQL and SQLite	Navicat.exe /backupprofile mysql ConnectionName DatabaseName ProfileName	
		Navicat.exe /backupprofile pgsql ConnectionName DatabaseName SchemaName ProfileName	
		Navicat.exe /backupprofile sqlite ConnectionName DatabaseName ProfileName	

Import	All	Navicat.exe /import mysql ConnectionName DatabaseName ProfileName	.npi
		Navicat.exe /import ora ConnectionName SchemaName ProfileName	.nopi
		Navicat.exe /import pgsq1 ConnectionName DatabaseName SchemaName ProfileName	.np1pi
		Navicat.exe /import sqlite ConnectionName DatabaseName ProfileName	.nlpi
		Navicat.exe /import mssql ConnectionName DatabaseName SchemaName ProfileName	.ns1pi
Export Table	All	Navicat.exe /export mysql ConnectionName DatabaseName ProfileName	.npe
		Navicat.exe /export ora ConnectionName SchemaName ProfileName	.nope
		Navicat.exe /export pgsq1 ConnectionName DatabaseName SchemaName ProfileName	.np1pe
		Navicat.exe /export sqlite ConnectionName DatabaseName ProfileName	.nlpe
		Navicat.exe /export mssql ConnectionName DatabaseName SchemaName ProfileName	.nspe
Export View Result	All	Navicat.exe /exportview mysql ConnectionName DatabaseName ProfileName	.npev
		Navicat.exe /exportview ora ConnectionName SchemaName ProfileName	.nopev
		Navicat.exe /exportview pgsq1 ConnectionName DatabaseName SchemaName ProfileName	.np1pev
		Navicat.exe /exportview sqlite ConnectionName DatabaseName ProfileName	.nlpev
		Navicat.exe /exportview mssql ConnectionName DatabaseName SchemaName ProfileName	.nspev
Export Materialized View Result	Oracle	Navicat.exe /exportmview ora ConnectionName SchemaName ProfileName	.nopem



Export Query Result	All	Navicat.exe /exportquery mysql ConnectionName DatabaseName ProfileName	.npeq
		Navicat.exe /exportquery ora ConnectionName SchemaName ProfileName	.nopeq
		Navicat.exe /exportquery pgsql ConnectionName DatabaseName SchemaName ProfileName	.nppeq
		Navicat.exe /exportquery sqlite ConnectionName DatabaseName ProfileName	.nlpeq
		Navicat.exe /exportquery mssql ConnectionName DatabaseName SchemaName ProfileName	.nspeq
Query Execution	All	Navicat.exe /query mysql ConnectionName DatabaseName QueryName	.sql
		Navicat.exe /query ora ConnectionName SchemaName QueryName	
		Navicat.exe /query pgsql ConnectionName DatabaseName SchemaName QueryName	
		Navicat.exe /query sqlite ConnectionName DatabaseName QueryName	
		Navicat.exe /query mssql ConnectionName DatabaseName SchemaName QueryName	
Data Transfer	All	Navicat.exe /datatransfer mysql ProfileName	.npt
		Navicat.exe /datatransfer ora ProfileName	.nopt
		Navicat.exe /datatransfer pgsql ProfileName	.nppt
		Navicat.exe /datatransfer sqlite ProfileName	.nlpt
		Navicat.exe /datatransfer mssql ProfileName	.nspt
		Navicat.exe /datatransfer premium ProfileName	.napt
Data Synchronization	All	Navicat.exe /datasync mysql ProfileName	.npd
		Navicat.exe /datasync ora ProfileName	.nopd
		Navicat.exe /datasync pgsql ProfileName	.npdp
		Navicat.exe /datasync sqlite ProfileName	.nlpd
		Navicat.exe /datasync mssql ProfileName	.nspdp
Batch Jobs	All	Navicat.exe /batchjob BatchJobName	.npj

Print Report	All	Navicat.exe /report mysql ConnectionName DatabaseName ReportName Param1 Param2	.rtm
		Navicat.exe /report ora ConnectionName SchemaName ReportName Param1 Param2	
		Navicat.exe /report pgsql ConnectionName DatabaseName SchemaName ReportName Param1 Param2	
		Navicat.exe /report sqlite ConnectionName DatabaseName ReportName Param1 Param2	
		Navicat.exe /report mssql ConnectionName DatabaseName SchemaName ReportName Param1 Param2	

**Note:**

Param1 - file type: pdf, htmlfile or excelfile

Param2 - printer name or target file path, e.g. C:\Users\user1\Desktop\test1.pdf

## Navicat for MySQL

Start Navicat from command line:

Navicat Objects	Command Lines	File Extensions
Backup	Navicat.exe /backup ConnectionName DatabaseName	compressed (.psc), uncompressed (.psb)
Backup Server	Navicat.exe /backupserver ConnectionName	
Backup Database	Navicat.exe /backupdatabase ConnectionName DatabaseName	
Backup Profile	Navicat.exe /backupprofile ConnectionName DatabaseName ProfileName	
Import	Navicat.exe /import ConnectionName DatabaseName ProfileName	.npi
Export Table	Navicat.exe /export ConnectionName DatabaseName ProfileName	.npe
Export View Result	Navicat.exe /exportview ConnectionName DatabaseName ProfileName	.npev
Export Query Result	Navicat.exe /exportquery ConnectionName DatabaseName ProfileName	.npeq
Query Execution	Navicat.exe /query ConnectionName DatabaseName QueryName	.sql
Data Transfer	Navicat.exe /datatransfer ProfileName	.npt
Data Synchronization	Navicat.exe /datasync ProfileName	.npd
Batch Jobs	Navicat.exe /batchjob BatchJobName	.npj
Print Report	Navicat.exe /report ConnectionName DatabaseName ReportName Param1 Param2	.rtm

### Note:

Param1 - file type: pdf, htmlfile or excelfile

Param2 - printer name or target file path, e.g. C:\Users\user1\Desktop\test1.pdf

## Navicat for Oracle

Start Navicat from command line:

Navicat Objects	Command Lines	File Extensions
Import	Navicat.exe /import ConnectionName SchemaName ProfileName	.nopi
Export Table	Navicat.exe /export ConnectionName SchemaName ProfileName	.nope
Export View Result	Navicat.exe /exportview ConnectionName SchemaName ProfileName	.nopev
Export Materialized View Result	Navicat.exe /exportmview ConnectionName SchemaName ProfileName	.nopem
Export Query Result	Navicat.exe /exportquery ConnectionName SchemaName ProfileName	.nopeq
Query Execution	Navicat.exe /query ConnectionName SchemaName QueryName	.sql
Data Transfer	Navicat.exe /datatransfer ProfileName	.nopt
Data Synchronization	Navicat.exe /datasync ProfileName	.nopd
Batch Jobs	Navicat.exe /batchjob BatchJobName	.nopj
Print Report	Navicat.exe /report ConnectionName SchemaName ReportName Param1 Param 2	.rtm

### Note:

Param1 - file type: pdf, htmlfile or excelfile

Param2 - printer name or target file path, e.g. C:\Users\user1\Desktop\test1.pdf

## Navicat for PostgreSQL

Start Navicat from command line:

Navicat Objects	Command Lines	File Extensions
Backup	Navicat.exe /backup ConnectionName DatabaseName SchemaName	compressed (.psc), uncompressed (.psb)
Backup Server	Navicat.exe /backupserver ConnectionName	
Backup Database	Navicat.exe /backupdatabase ConnectionName DatabaseName	
Backup Profile	Navicat.exe /backupprofile ConnectionName DatabaseName SchemaName ProfileName	
Import	Navicat.exe /import ConnectionName DatabaseName SchemaName ProfileName	.nppi
Export Table	Navicat.exe /export ConnectionName DatabaseName SchemaName ProfileName	.nppe
Export View Result	Navicat.exe /exportview ConnectionName DatabaseName SchemaName ProfileName	.nppev
Export Query Result	Navicat.exe /exportquery ConnectionName DatabaseName SchemaName ProfileName	.nppeq
Query Execution	Navicat.exe /query ConnectionName DatabaseName SchemaName QueryName	.sql
Data Transfer	Navicat.exe /datatransfer ProfileName	.nppt
Data Synchronization	Navicat.exe /datasync ProfileName	.nppd
Batch Jobs	Navicat.exe /batchjob BatchJobName	.nppj
Print Report	Navicat.exe /report ConnectionName DatabaseName SchemaName ReportName Param1 Param2	.rtm

### Note:

Param1 - file type: pdf, htmlfile or excelfile

Param2 - printer name or target file path, e.g. C:\Users\user1\Desktop\test1.pdf

**Hint:** You can specify the schema for backup. If you omit the schema name, backup will run for the whole database which includes all schemas.

## Navicat for SQLite

Start Navicat from command line:

Navicat Objects	Command Lines	File Extensions
Backup	Navicat.exe /backup ConnectionName DatabaseName	compressed (.psc), uncompressed (.psb)
Backup Server	Navicat.exe /backupserver ConnectionName	
Backup Database	Navicat.exe /backupdatabase ConnectionName DatabaseName	
Backup Profile	Navicat.exe /backupprofile ConnectionName DatabaseName ProfileName	
Import	Navicat.exe /import ConnectionName DatabaseName ProfileName	.nlpi
Export Table	Navicat.exe /export ConnectionName DatabaseName ProfileName	.nlpe
Export View Result	Navicat.exe /exportview ConnectionName DatabaseName ProfileName	.nlpev
Export Query Result	Navicat.exe /exportquery ConnectionName DatabaseName ProfileName	.nlpeq
Query Execution	Navicat.exe /query ConnectionName DatabaseName QueryName	.sql
Data Transfer	Navicat.exe /datatransfer ProfileName	.nlpt
Data Synchronization	Navicat.exe /datasync ProfileName	.nlpd
Batch Jobs	Navicat.exe /batchjob BatchJobName	.nlpj
Print Report	Navicat.exe /report ConnectionName DatabaseName ReportName Param1 Param2	.rtm

### Note:

Param1 - file type: pdf, htmlfile or excelfile

Param2 - printer name or target file path, e.g. C:\Users\user1\Desktop\test1.pdf

## Navicat for SQL Server

Start Navicat from command line:

Navicat Objects	Command Lines	File Extensions
Import	Navicat.exe /import ConnectionName DatabaseName SchemaName ProfileName	.nspi
Export Table	Navicat.exe /export ConnectionName DatabaseName SchemaName ProfileName	.nspe
Export View Result	Navicat.exe /exportview ConnectionName DatabaseName SchemaName ProfileName	.nspev
Export Query Result	Navicat.exe /exportquery ConnectionName DatabaseName SchemaName ProfileName	.nspeq
Query Execution	Navicat.exe /query ConnectionName DatabaseName SchemaName QueryName	.sql
Data Transfer	Navicat.exe /datatransfer ProfileName	.nspt
Data Synchronization	Navicat.exe /datasync ProfileName	.nspd
Batch Jobs	Navicat.exe /batchjob BatchJobName	.nspj
Print Report	Navicat.exe /report ConnectionName DatabaseName SchemaName ReportName Param1 Param2	.rtm

### Note:

Param1 - file type: pdf, htmlfile or excelfile

Param2 - printer name or target file path, e.g. C:\Users\user1\Desktop\test1.pdf

## Navicat Support Information

### Navicat Support Center

URL : [http://www.navicat.com/en/support/support\\_ticket/submit\\_ticket.html](http://www.navicat.com/en/support/support_ticket/submit_ticket.html)

1. Navicat Wiki - Browse the Navicat Wiki for frequently asked questions
2. Documentation - View online manuals and articles, download PDF manuals.
3. Support Ticket - Submit a trouble ticket to a department, check current ticket status
4. Live Support - Chat with our staff
5. Survey - Tell us your comments on Navicat

### How To Upgrade Navicat or Retrieve download information again

If you have purchased our products and want to receive the upgrade information of the latest version in future, please visit at our [Customer Center](#).

### Report bugs

If you feel you have encountered a bug in your PremiumSoft product, please notify us.

Bug reports are defined as:

1. Any unexpected error encountered that can consistently be re-created
2. Any action that causes your product or system to freeze up

To submit your bug report, please contact our support team via Navicat Support Center.

URL : [http://www.navicat.com/en/support/support\\_ticket/submit\\_ticket.html](http://www.navicat.com/en/support/support_ticket/submit_ticket.html)





# ОГНЕЛИС

Покупай  
лицензионное  
программное обеспечение  
со скидкой

---

Buy  
legal  
software  
with a discount

[www.ognelis.ru](http://www.ognelis.ru)

